

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

PREDIKCE DEŠTĚ POMOCÍ METEOROLOGICKÉHO RADARU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETR GERYCH

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

PREDIKCE DEŠTĚ POMOCÍ METEOROLOGICKÉHO RADARU

RAIN PREDICTION USING METEO-RADAR

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR GERYCH

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IGOR SZÖKE

BRNO 2009

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2008/2009

Zadání bakalářské práce

Řešitel: **Gerych Petr**

Obor: Informační technologie

Téma: **Predikce deště pomocí meteorologického radaru**

Kategorie: Počítačová grafika

Pokyny:

1. Seznamte se s teorií detekce směru pohybu v obraze a s jeho predikcí
2. Implementujte systém krátkodobé predikce (cca +2 hodiny) srážek za použití dat z meteorologického radaru (např. <http://www.chmi.cz/meteo/rad/index.html>)
3. Ověřte funkčnost aplikace a určete přesnost predikce.

Literatura:

- Dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Pro zápočet v zimním semestru je třeba splnit bod 1 a alespoň část bodu 2 ze zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

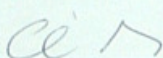
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Szőke Igor, Ing.,** UPGM FIT VUT

Datum zadání: 1. listopadu 2008

Datum odevzdání: 20. května 2009

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 01 S. no. Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Práce se zabývá krátkodobou predikcí deště. Jako vstupní data jsou využívány statické snímky z meteorologických radarů. Je zde popsán princip práce těchto radarů. Možné metody pro detekci objektu, registraci objektu, interpolace a extrapolace pohybu. Pro predikci deště se využívá algoritmus pro semínkové vyplňování a extrapolace pomocí Lagrangeova interpolačního polynomu. Aplikace je napsána v jazyce C++ pod OS Linux. Jsou zde také uvedeny příklady výsledků aplikace.

Abstract

This paper deal with the rain prediction in the short time interval. The static pictures from meteo-radar serves as input data. The principle of meteo-radar is explained. The possible methods of the object detection and registration, motion interpolation and extrapolation is described. The flood fill algorithm and Lagrange extrapolation is applied to rain prediction. Application is written in C++ language under OS Linux. The example of the software application results is included.

Klíčová slova

predikce, déšť, detekce pohybu v obraze, Lagrange, interpolace, extrapolace, semínkové vyplňování, detekce objektů, registrace objektů

Keywords

prediction, rain, motion detection in picture, Lagrange, interpolation, extrapolation, flood fill, objects detection, objects registration

Citace

Petr Gerych: Predikce deště pomocí meteorologického radaru, bakalářská práce, Brno, FIT VUT v Brně, 2009

Predikce deště pomocí meteorologického radaru

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Igora Szöke

.....

Petr Gerych
16. května 2009

Poděkování

Chtěl bych poděkovat mému vedoucímu bakalářské práce za pomocné rady.

© Petr Gerych, 2009.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	3
2 Způsoby predikce v meteorologii	4
2.1 Meteorologická teorie	4
2.1.1 Využití radarů v meteorologii	4
2.1.2 Princip radaru	5
2.1.3 Zobrazování radarových dat	6
2.1.4 Meteorologická interpretace radarových měření	6
2.1.5 Formát dat	7
2.2 Využívané metody pro predikci	10
3 Zvolený postup řešení predikce	11
4 Detekce a extrakce objektů	13
4.1 Tvar objektů	13
4.2 Princip semínkového algoritmu	15
4.2.1 Algoritmus Scan-line seed fill	17
4.2.2 Shrnutí	17
5 Registrace mezi statickými snímky	18
5.1 Filtrace rušivých objektů	18
5.2 Výpočet těžiště	18
5.3 Problémy mizení a objevování objektů	19
5.4 Slučování do větších celků	19
5.5 Přiřazení mezi jednotlivými snímky	20
6 Predikce deště	22
6.1 Lineární predikce	22
6.2 Nelineární predikce	23
6.3 Interpolace algebraickými polynomy	23
6.3.1 Princip interpolace	24
6.3.2 Důkaz	25
6.3.3 Známé metody	25
6.4 Lagrangeův interpolační polynom	25
6.4.1 Konstrukce interpolačního polynomu	26
6.4.2 Příklad	27
6.5 Extrapolace využívající Lagrangeovu metodu	28
6.5.1 Problém vracení se objektů	28

6.5.2	Nejpřesnější stupeň polynomu	31
7	Určení polohy predikovaného místa	32
7.1	Souřadnice GPS	33
8	Vyhodnocení predikce	34
8.1	Testování na snímcích	34
8.2	Porovnání řešení	34
9	Závěr	37
9.1	Návrhy rozšíření	37
A	Manuál	40
B	Konfigurační soubor	43

Kapitola 1

Úvod

Práce poskytuje náhled do problematiky predikování srážkových jevů v meteorologii. Vybrána byla metoda, která předpokládá v souladu se zadáním krátkodobou predikci a je založena na detekci a sledování objektů. Vstupní informace jsou k dispozici ve formě statických snímků, které poskytl pro účely bakalářské práce Český hydrometeorologický ústav (ČHMÚ [7]).

První část práce vysvětluje základní používané prostředky v meteorologii, způsob interpretace jejich dat a různé metody predikce přičemž jedna z těchto metod byla použita jako výchozí.

Další část je věnována detekování a extrahování jednotlivých objektů ze statických snímků a s tímto spojenými problémy.

V třetí části se popisuje implementovaný způsob spárování extrahovaných objektů mezi jednotlivými snímky, kde se řeší také problémy jako vyfiltrování rušivých objektů, problémy mizení a objevování atd.

Čtvrtá část již rozebírá možné způsoby predikce na základě získaných informací a zaměřuje se hlavně na použití Lagrangeova interpolačního polynomu.

Předposlední pátá část práce popisuje řešení identifikace polohy predikovaného místa pomocí souřadnicového systému GPS.

Šestá poslední část vyhodnocuje dosažené výsledky, návrhy možných rozšíření do budoucnosti, pro které je aplikace uzpůsobena. Také je zde uvedeno, dle pokynů našeho společného vedoucího bakalářské práce pana Ing. Igora Szöke, porovnání s implementací stejného zadání bakalářské práce, kterou vypracoval kolega Lukáš Putna. S kterým jsme se domluvily na společném testsetu a metrikách pro porovnání úspěšnosti aplikace.

Kapitola 2

Způsoby predikce v meteorologii

ČHMÚ provádí predikci počasí numerickým předpovědním modelem. Řeší se soustava rovnic popisující fyzikální zákony. Data pro výpočty se získávají z meteorologických měření (přízemní observatoře, balónové sondy, meteorologické radary a družice). K výpočtům se používá vysoce výkonný a specializovaný hardware.

V práci se řeší predikce deště na základě informací ze statických snímků, které obsahují data z meteorologických radarů.

2.1 Meteorologická teorie

Hlavní využití v meteorologii mají radiolokátory (met. radary), které jsou používány k určení rozložení okamžitých intenzit atmosférických srážek většinou na ploše, která se pohybuje v řádech 100 000 km. Princip jakým monitorují tyto srážky je odvozen od schopnosti odrazet mikrovlny od srážkových částic v atmosféře jako jsou například vodní kapičky, sněhové vločky, atd.

2.1.1 Využití radarů v meteorologii

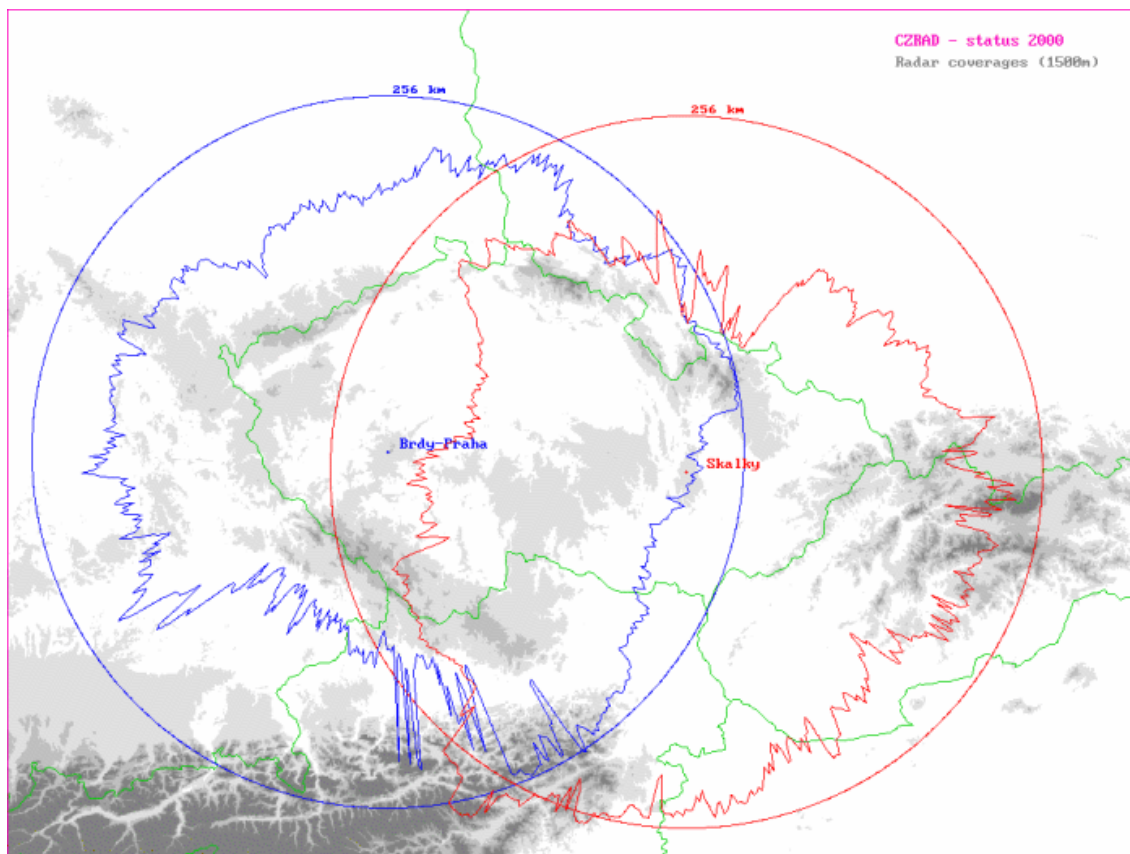
Dále se používá síť pozemních stanic a družicová pozorování pro synoptickou a leteckou meteorologii. Tyto prostředky jsou dobře doplňovány právě radiolokátory, které mají dobré plošné pokrytí i dobré prostorové a časové rozlišení dat.

Tímto způsobem je umožněno určovat velmi krátkodobou předpověď (v řádech stovek minut) a varování před nebezpečnými jevy, spojenými s konvektivní oblačností jako jsou například silné kroupy, významnější kroupy, atd.

Radary, které se využívají v meteorologii většinou mívají účinný dosah pro určování intenzit srážek přibližně 100 - 150 km. Větší dosah mívají pro rozpoznání větší bouřkové oblačnosti a to přibližně do 250 - 300 km.

Tyto účinné dosahy jsou omezovány vzdáleností ve které je šíření paprsku utlumováno vlivem zakřivení zemského povrchu. Částečně jdou tyto vlivy eliminovat kombinovat zapojením do měření více radarů, které se propojí do tzv. radarové sítě.

Takové radarové sítě se tvořily v průběhu 70. a 80. let. Později v 90. letech pozvolna vznikala mezinárodní výměna operativních radarových dat. Jako je například středoevropská radarová síť (CERAD). Meteorologické radary pokrývají území České Republiky jak je zobrazeno na obrázku 2.1. Jsou zde pouze dva radary, které ale dostačují pro pokrytí celé ČR.



Obrázek 2.1: Pokrytí území ČR radary ČHMÚ

2.1.2 Princip radaru

Radar pracuje na principu vysílání krátkých pulsů s vysokým okamžitým výkonem v řádech 100 kW. Tyto pulsy jsou vysílány parabolickou anténou o průměru několika metrů, který má tvar úzkého svazku. Jakmile jsou vyslány do atmosféry, tak se část energie odrazí zpět od cílů meteorologických i nemeteorologických (srážkové částice, letadla, atd.). Tato část odražené energie je přijata parabolickou anténou, zesílena a detekována přijímačem radaru.

Poloha cíle se určí pomocí polohy antény (přesněji jejího azimutu a elevace) a času mezi vysláním a příjmem signálu. Intenzita cíle se určí podle síly odražené energie. Síla odražené energie je přímo úměrná radiolokační odrazivosti Z cíle. Taková odrazivost se dá určit pomocí rovnice 2.1.

$$Pr = PI * (Z/R^2) \quad (2.1)$$

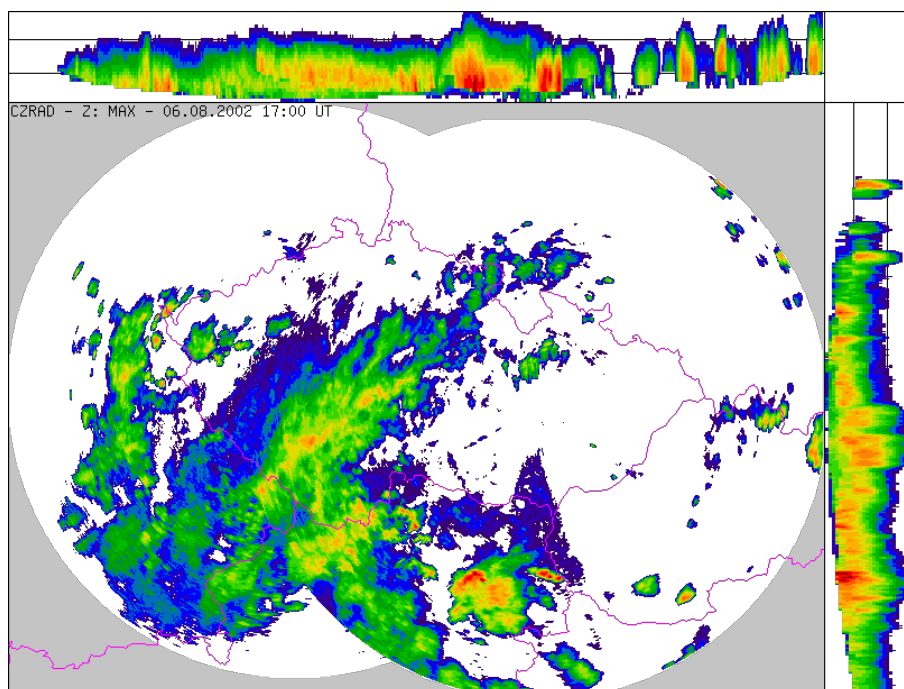
, kde :

- Pr - přijatý výkon [kW]
- PI - meteorologický potenciál radaru (přístrojová konstanta) [kW]
- R - vzdálenost cíle [m]

Pro meteorologické cíle, skládající se z velkého množství vodních a ledových částic, je radiolokační odrazivost Z přímo úměrná sumě 6. mocnin průměrů částic v jednotkovém objemu ($Z = \sum D^6$). To platí přesně pro částice s průměrem řádově menším než vlnová délka radaru. Jednotkou odrazivosti je $1 \text{ mm}^6/\text{m}^3$, pro praktické účely se užívá logaritmická jednotka dBZ, kde $Z[\text{dBZ}] = 10 \cdot \log(Z[\text{mm}^6/\text{m}^3])$, tedy 0 dBZ odpovídá $Z = 1 \text{ mm}^6/\text{m}^3$.

2.1.3 Zobrazování radarových dat

Data jsou zobrazována na počítačovém monitoru v barevné stupnici intenzit (obr. 2.2). Obvyklý interval měření je 10 - 15 min. Horizontální rozlišení dat bývá $2 \times 2 \text{ km}$, vertikální 1 km. Pokud je dosah radaru nejčastějších 250 km, tak se celý obraz skládá z přibližně 250×250 obrazových pixelů. Toto prostorové rozlišení je potřebné, aby bylo možné zachytit jednotlivá srážková jádra přeháněk.



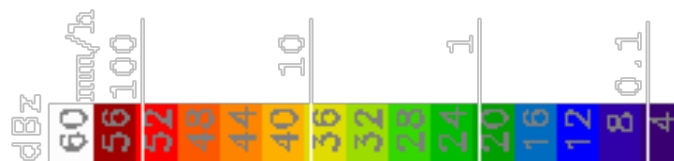
Obrázek 2.2: Příklad zobrazení dat

2.1.4 Meteorologická interpretace radarových měření

Radarová měření se zaznamenávají ve formě barevných pixelů. Tyto pixely mohou nabývat 15 různých barev z tzv. barevné stupnice odrazivosti (viz. 2.3). Krajní barvy ukazují hladiny končící vždy sudým kilometrem až do výšky 14km. Barvy končící kilometrem dělitelným čtyřmi (4, 8 a 12 km) jsou označeny zelenými odstíny, hladina do 10 km je označena pak červeně. Každý stupeň odrazivosti je počítán s krokem 4 dBZ. Tato hodnota kroku odpovídá intenzitě deště přibližně 0.06 mm/h.

Pro přibližný přepočít odrazivosti na intenzity srážek platí exponenciální závislost :

$Z[\text{dBZ}]$	7	23	39	55
$I[\text{mm/h}]$	0.1	1	10	100



Obrázek 2.3: Barevná stupnice odrazivosti

Některé hodnoty jsou ve výstupním obrazu dostatečně eliminovány. Lze je rozpoznat například pomocí jejich dopplerovské rychlosti, která se blíží nule nebo-li stojí na místě.

Pro meteorologické účely můžeme dělit objekty pozorované radarem na :

1. *Cíle meteorologické*, z nichž lze rozlišovat cíle konvektivního a vrstevnatého charakteru. Pro které platí hlavní vlastnosti jako jsou např. rozsah obvykle větší než jednotlivé pixely, spíše kompaktní tvar, časové změny odrazivosti jsou malé, na animaci se jeví zřetelný pohyb, objevují se a mizí postupně.
2. *Cíle ostatní*, obvykle rušivé, z nichž nejvýznamnější jsou pozemní cíle (odrazy od terénu). Tyto cíle se vyznačují hlavními vlastnostmi jako jsou jednotlivé pixely s vyšší odrazivostí nebo menší nesouvislé oblasti, velká časová proměnlivost (někdy též náhlé objevení se nebo zmizení v závislosti na šíření paprsku), zvýšení odrazivosti po srážkách, izolované body ve větších výškách (odrazy od letadel), vlastní šum a nestabilita vysílače radaru, rušení radaru jiným signálem na blízké frekvenci.

Bližší informace o meteorologické teorii lze nalézt na stránkách Českého hydrometeorologického institutu [7].

2.1.5 Formát dat

Data srážek nad Českou Republikou poskytl Český hydrometeorologický ústav. Výběr dat provedl na základě svých zkušeností ČHMÚ. Vybraná data obsahují extrémní události srážkových jevů. Jsou v nich obsaženy roky 2002, 2004, 2005 a 2006.

Tento formát mi byl poskytnut vždy z určitého data pro celý den. Časové intervaly mezi jednotlivými statickými snímky jsou 10 minut. Z čehož vyplývá, že připadá pro jedno datum (den) 144 statických snímků.

Pozn.: Na oficiálním webu CHMI, kde lze sledovat aktuální radarové snímky [5] se tyto snímky publikují po 30 minutách.

Čistá data, která poskytnou radarové měření nejsou pro počítačové zpracování vhodná, proto se musejí přepočítat (převést) do přívětivějšího formátu. Tento převod je výpočetně náročný, proto byl proveden také ČHMÚ, který má potřebné technické a odborné znalosti.

ČHMÚ poskytl data, ve dvou formátech RPD a PNG.

Formát RPD

Tento formát se používá pro radarová měření. Jeho hlavní vlastnosti jsou, že je univerzální a nezávislý na výrobci radaru. Data jsou ukládána v kartézských souřadnicích.

Jak je popsáno v dokumentaci formátu RPD soubor se skládá ze dvou částí:

- Textové hlavičky s řádky obsahujícími dvojice klíčových slov a hodnot popisujících základní parametry objemového měření.
- Binárních dat. Radarová data jsou uložena ihned za textovou hlavičkou.

Konvence pojmenovávání RPD souborů:

`<georeference>.<název product>.<datum>.<čas>.<délka akumulace>.rpd`

symbolicky: `REFER.PROD.YYYYMMDD.hhmm.DTIME.rpd`

Formát PNG

Formát PNG je používán pro bezztrátovou kompresi rastrové grafiky. Hlavní využití mívá na internetu.

Tento formát se také skládá ze dvou hlavních částí:

- Hlavičky souboru PNG, která má délku 8 bytů. Tyto byty mají vždy stejné hodnoty (hexadecimálně: 89 50 4e 47 0d 0a 1a 0a)
- Těla, které je složeno z několika chunků. Chunky se skládají z hlavičky (čtyři ASCII znaky) určené délky a typu. Dělí se na rozhodující (první znak velké písmeno) a pomocné (první znak malé písmeno). Dále následují aktuální data.

Vybral jsem si formát dat PNG, který je standardnější pro práci s počítačovou grafikou. Pro zpracování tohoto formátu jsem použil open source knihovnu libpng [\[3\]](#) pro jazyk C.

Využívané geografické projekce (georeference)

Geografické projekce se používají k vytvoření výše uvedených formátů z radarových dat. Jako základní se používá gnómonická projekce se středem v Praze-Libuši. Tato projekce se dříve využívala u starších horizontálních rozlišení 2x2 km, ale nachází uplatnění i u novějších rozlišení 1x1 km. Do projekce jsou konvertovány jednotlivé produkty přímo z objemových dat. Projekce se vytvoří pro jednotlivé radary a z nich se vytvoří sloučená informace posunutím ve směru x a y.

Od ČHMÚ mi byla poskytnuta projekce pacz23. Tato projekce je sloučená informace z radarů Praha-Brdy (pacz53) a Skalky (pacz63) s rozlišením 1x1 km.

- Parametry projekce pacz53:

```
proj=gnom
lat_0=50.008, lon_0=14.447 m,
a=6379000
x_0=86100 m, y_0=-201700 m, es=0.no_defs <>

x_res = 512 # počet sloupců
y_res = 512 # počet řádků
pix_res = 1.0 # velikost pixelu ve středu projekce v~[km]
```

- Parametry projekce pacz63:

```
proj=gnom
lat_0=50.008, lon_0=14.447,
a=6379000.
x_0=300800 m, y_0=-216700 m, es=0.no_defs <>

x_res = 512 # počet sloupců
y_res = 512 # počet řádků
pix_res = 1.0 # velikost pixelu ve středu projekce v~[km]
```

- Parametry sloučené informace pacz23:

```
proj=gnom
lat_0=50.008, lon_0=14.447
a=6379000
x_0=301500 m, y_0=-217500 m, es=0.no_defs <>

x_res = 728 # počet sloupců
y_res = 528 # počet řádků
pix_res = 1.0 # velikost pixelu ve středu projekce v~[km]
```

Pro vygenerování sloučené informace (pacz23) z jednotlivých radarů (pacz53, pacz63) platí následující hodnoty posunutí:

	x-posun [px]	y-posun [px]	radar
pacz53	215	15	skalky
pacz63	5	10	brdy

Parametry ostatních používaných projekcí jsou popsány na webu pro GPL software PROJ4 [6].

2.2 Využívané metody pro predikci

Z popsané teorie radarových měření vyplývají jisté vlastnosti, které musíme brát v úvahu, abychom byli schopni co nejpřesněji predikovat pozici a intenzitu srážkových oblačností. Jedná se například o přístup neboli nahlížení na statické snímky a od nich se odvíjejících problémů.

Zkoumal jsem metody, kterými by se dala predikce provádět. Dospěl jsem ke třem metodám:

1. Lineární predikce
2. Nelineární predikce
3. Neuronové sítě

První dvě metody pracují s pozorování objektů. Proto se ve statických snímcích musejí provádět detekce objektů, registrace objektů mezi statickými snímky, atd.

Třetí metoda, která využívá neuronové sítě je na snímky nahlíženo jako na celek a proto se nemusejí detekovat žádné objekty. Toto ulehčení je kompenzováno dosti složitými principy práce neuronových sítí.

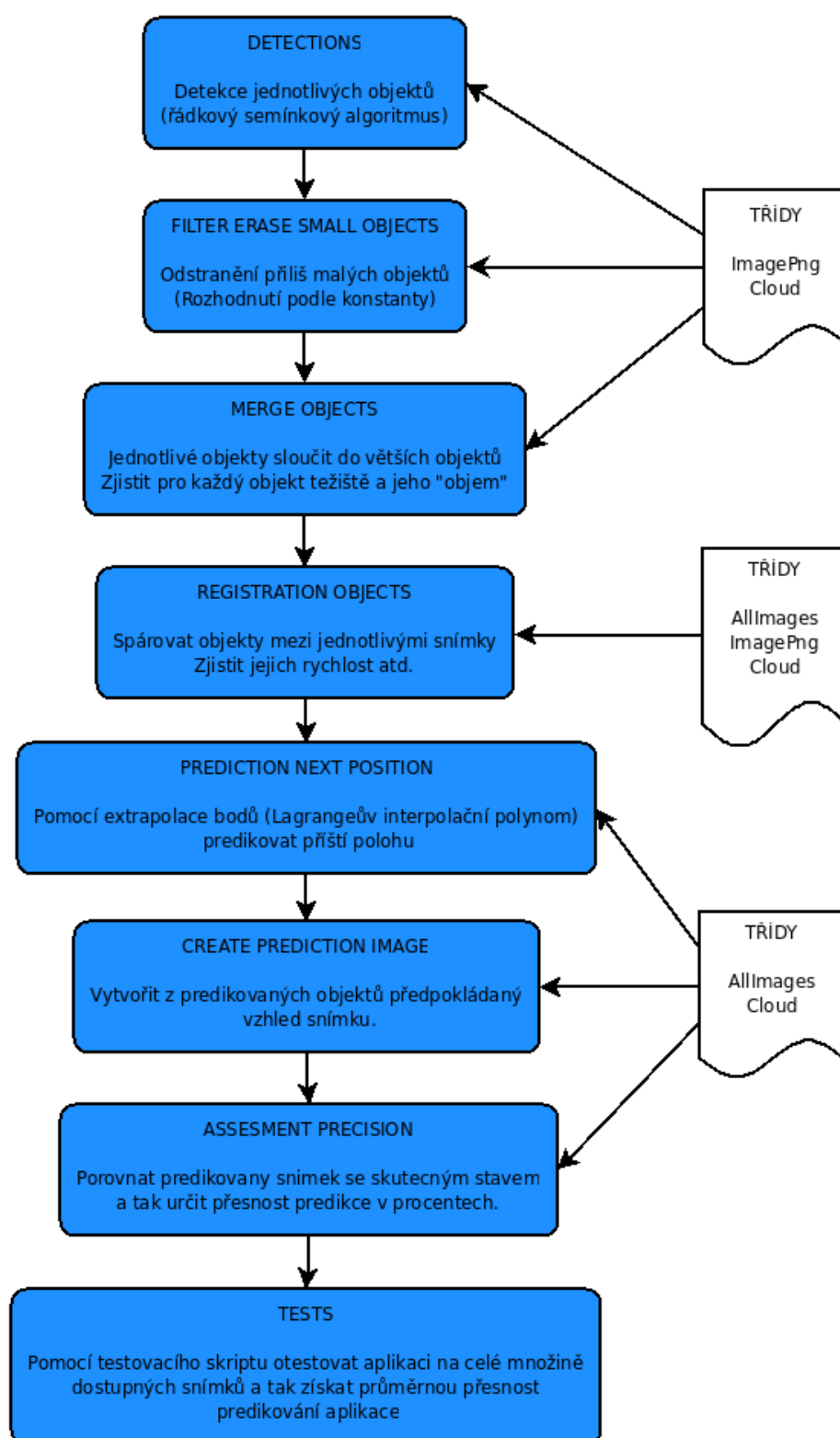
Kapitola 3

Zvolený postup řešení predikce

Úkolem bylo vytvořit systém, který dokáže provádět krátkodobou predikci na základě reálných dat meteorologického radaru (např. ČHMI [5]). Lineární predikce není vhodná z důvodu náhodného směru pohybu oblačností. Neuronové sítě jsou vysoce náročná operace při jejich trénování a vyžadují nadstandardní hardwarové vybavení. Nelineární predikce potlačuje problém náhodného směru pohybu oblačnosti při malém požadavku na početní výkon a mohla by vykazovat srovnatelné výsledky s neuronovými sítěmi, proto jsem si tuto metodu vybral a implementoval v jazyce C++.

Implementaci jsem provedl podle vývojového diagramu 3.1.

Vývojový diagram pro predikci srážek



Obrázek 3.1: Vývojový diagram aplikace

Kapitola 4

Detekce a extrakce objektů

Potřebná meteorologická data máme již k dispozici. Otázkou je jak tyto data co nejefektivněji zpracovat, aby bylo možné predikovat srážky nad Českou republikou.

Na statický snímek se dá nahlížet v podstatě dvěma způsoby:

- Rozlišovat a detekovat jednotlivé objekty na statickém obrázku a ty dále sledovat a analyzovat.
- Brát snímek jako celek (zkoumat pixel po pixelu), kde by se nemusely rozpoznávat jednotlivé objekty.

Extrakce relevantních objektů (oblačností) ze statického obrazu lze provádět více způsoby:

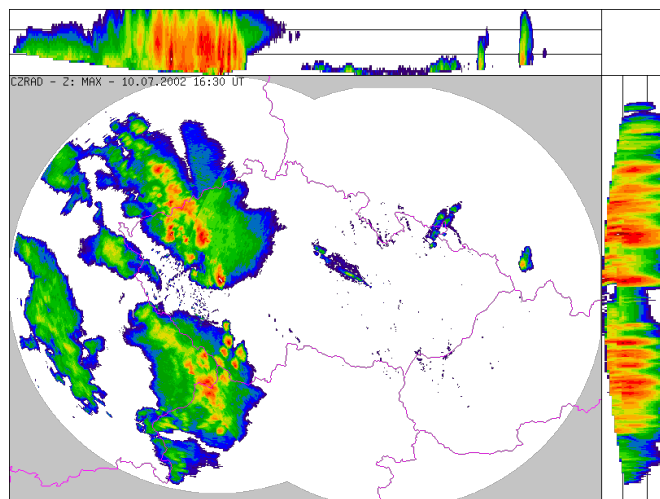
- Metody, které vyhodnocují tvar nebo velikost objektu (např. člověk - většinou tvar eliptického tvaru).
- Rozpoznávání s jakou pravděpodobností patří daný pixel danému objektu nebo pozadí (např. pomocí směsí Gaussových rozložení pravděpodobností).
- Rozhodnutí jest-li daný pixel patří objektu nebo pozadí, kdy je nám předem znám interval barev, kterých může objekt nabývat (např. algoritmus semínkového vyplňování).

Pro predikci deště jsem se rozhodl rozlišovat a detekovat jednotlivé objekty (z důvodů uvedených v sekci 4.1). K extrakci těchto objektů využít princip Semínkového vyplňování.

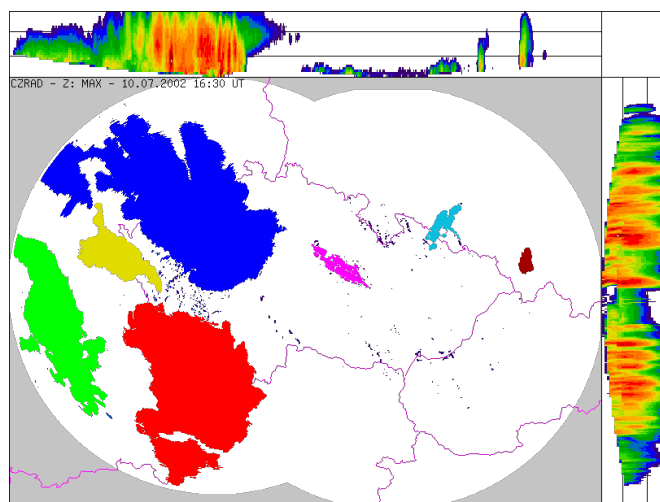
4.1 Tvar objektů

Při detekování objektů ve statickém snímku je nutné znát určité specifické vlastnosti hledaného objektu, abychom tento objekt mohli detekovat a dále sledovat. Nabízí se nejjednodušší popsání libovolného objektu a to podle určitého celistvého tvaru (obdélník, kruh, elipsa, mnohoúhelník, atd.). Bohužel takový způsob není pro případ srážek vhodný, protože mraky netvoří tyto tvary (snad jenom v naprosto vzácných situacích), ale nabývají úplně náhodných nepravidelných tvarů, jak je vidět například na obrázku 4.2.

Také podle dalších vlastností jako jsou části, které by měl objekt obsahovat nebo vlastnosti, které by měl splňovat nemůžeme mrak popsat.



Obrázek 4.1: Objekty se všemi barvami



Obrázek 4.2: Barevně zvýrazněné tvary jednotlivých objektů

Na statických snímcích z radaru mohou mraky zahrnovat libovolné barvy z palety barev, která se skládá z patnácti částí (viz. 4.1). Takže nepřichází v úvahu ani rozpoznání objektů podle jedné barvy, ale musí se uvažovat všech patnáct barev. Díky vlastnosti, že snímek používá těchto patnáct barev čistě k vykreslování srážek a nikoliv k vykreslování pomocných věcí (např. hranice, pozadí, konce dosahu radarů, atd.), tak můžeme popsat objekt (mrak) pomocí těchto patnácti barev.

Byl vybrán algoritmus, který dokáže rozpoznat všechny pixely, které na sebe navazují hranami a jsou uvnitř intervalu povolených barev. Takový algoritmus se jmenuje semínkové vyplňování (Flood fill).

4.2 Princip semínkového algoritmu

Semínkový algoritmus (semínkové vyplňování) je prostředek, který se vyvinul prioritně pro obarvování (vyplňování) oblastí v rastru. Rastrová oblast, která má být vyplněna musí být popsána určitým způsobem spojitých bodů. Body musejí splňovat předpoklady pro tento algoritmus jako je informace o jejich barvě. Ta musí být v množině stejných, podobných bodů pro utváření hranic nebo oblastí.

Takové hraniční entity nemáme předem přesně definovány, proto nemůžeme jednoznačně rozhodnout co je uvnitř a co vně určité vyplňované oblasti. Z těchto důvodů se musí určitým způsobem (uživatel, náhodně, podle kritérií) určit na počátku počáteční pixel (semínko). Takto přesně definujeme vnitřek oblasti a může začít šíření semínka.

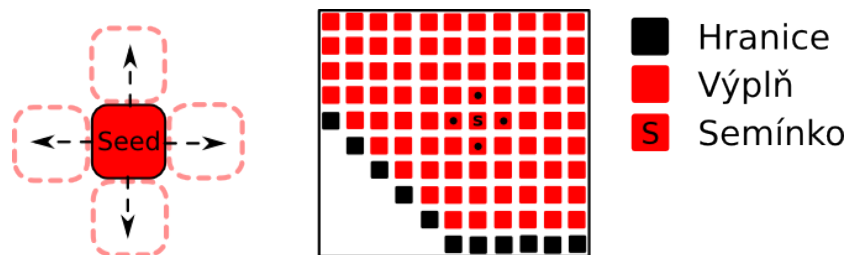
Šíření semínka může probíhat dvěma způsoby, kterým se říká 4 nebo 8 okolí. Pixely, které již podstoupily proces obarvení se automaticky stanou novými semínky. Šíření se aplikuje pouze na pixely dané barvy nebo daného intervalu barev. Většinou se při tom využívá rekurze.

Proces vyplňování můžeme rozdělit na čtyři různé varianty podle definice hranice nebo oblasti a to následovně:

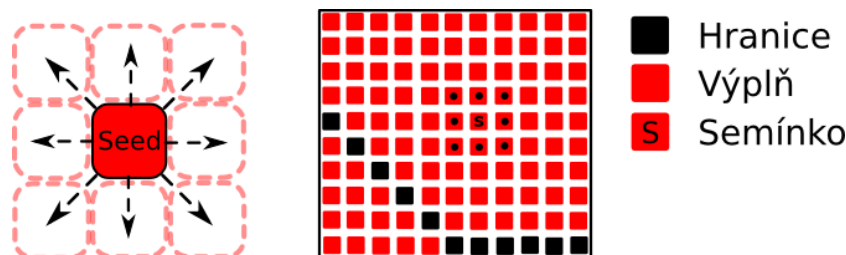
- Hraniční - oblast je definována spojitou hranicí z pixelů dané barvy. Při vyplňování se pixely testují, jestli nemají barvu hranice.
- Záplavové - oblast je definována spojitou množinou vnitřních pixelů dané barvy. Při vyplňování se pixely testují, jestli mají barvu oblasti.
- Měkké - oblast je definována spojitou hranicí z pixelů daného rozptylu barev. Při vyplňování se pixely testují, jestli nemají barvu uvnitř daného rozptylu.
- Prahové - oblast je definována spojitou množinou vnitřních pixelů jejichž barva leží v daném rozmezí okolo barvy semínka. Při vyplňování se pixely testují, jestli mají barvu uvnitř rozmezí okolo barvy semínka.

Jak se šíří barva semínka nebo jak sousedí pixely v definované hranici taky můžeme rozdělit na dva níže uvedené případy:

- 4-okolí - každý pixel má 4 sousedy, po hraně (viz. 4.3). Při vyplňování testujeme a obarvujeme tyto 4 sousedy. Hranice oblasti je uzavřená, pokud všechny její hraniční pixely sousedí po hraně.
- 8-okolí - každý pixel má 8 sousedů, po hraně a přes vrcholy (viz. 4.4). Při vyplňování testujeme a obarvujeme těchto 8 sousedů. Hranice oblasti je uzavřená, pokud všechny její hraniční pixely sousedí po hraně nebo přes vrchol.



Obrázek 4.3: Vyplňování typu 4-okolí



Obrázek 4.4: Vyplňování typu 8-okolí

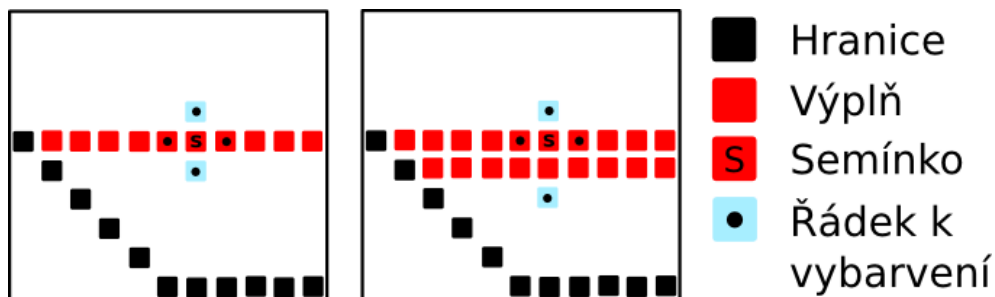
Semínkový vyplňovací algoritmus je možné velmi jednoduše implementovat (jak již bylo popsáno) pomocí rekurze. Zde však hrozí nebezpečí přetečení zásobníku při vyplňování rozsáhlejších oblastí. Proto je vhodnější implementovat algoritmus s ukládáním nových semínek do vlastního paměťového zásobníku.

Obecný postup vyplňování dané rastrové oblasti semínkovým algoritmem s vlastním zásobníkem pro semínko je pak následující:

1. Testování sousedních pixelů aktuálního semínka (4 nebo 8 podle typu vyplňování), jestli jsou ještě neobarvené a patří do dané oblasti.
2. Obarvení vyhovujících sousedních pixelů a jejich vložení do zásobníku.
3. Získání nového semínka z vrcholu zásobníku a návrat k bodu 1.
4. Konec, pokud je zásobník prázdný.

Tento algoritmus také není ideální, protože jeho paměťové nároky pro větší rastrové oblasti mohou být příliš náročné. Proto se používá modifikovaný algoritmus (řádkové semínkové vyplňování viz. 4.5), který má výhody ve snížení počtu přístupů do obrazové paměti a používání malého zásobníku.

Obecný postup vyplňování dané rastrové oblasti řádkovým semínkovým algoritmem s vlastním zásobníkem pro semínko je pak následující:



Obrázek 4.5: Scan-line seed fill

4.2.1 Algoritmus Scan-line seed fill

- Vložení semínka do zásobníku
- Dokud není zásobník prázdný:
 1. Vyjmutí semínka $[x, y]$ ze zásobníku
 2. Nalezení hranice x_L a x_R na řádku y v nejbližším okolí bodu $[x, y]$.
 3. Nakreslení úsečky $[x_L, y] - [x_R, y]$
 4. Hledání souvislých vnitřních úseků na (vyšší) úsečce $[x_L, y - 1] - [x_R, y - 1]$
 - každý nalezený úsek \Rightarrow vložení jednoho vnitřního bodu úseku do zásobníku
 5. Hledání souvislých vnitřních úseků na (nižší) úsečce $[x_L, y + 1] - [x_R, y + 1]$
 - každý nalezený úsek \Rightarrow vložení jednoho vnitřního bodu úseku do zásobníku

4.2.2 Shrnutí

K řízení procesu vyplňování používám prahovou metodu a k výběru sousedů 4-okolí. Jako nejefektivnější algoritmus jsem tedy použil řádkový semínkový algoritmus (Scan-line seed fill) s vlastním zásobníkem.

Kapitola 5

Registrace mezi statickými snímky

Po úspěšném získání jednotlivých objektů (mraků) musíme provést registraci mezi statickými snímky, abychom mohli sledovat trajektorie drah putování mraků. Pro co nejlepší spárování jednotlivých mraku musíme provést a získat určité parametry, které budou rozhodovat zda-li se jedná o identický mrak.

5.1 Filtrace rušivých objektů

Protože data od radaru nejsou ideální a obsahují předměty, které nepředstavují srážky, ale odrazy od jiných elementů jako například odrazy od letadel atd. (popsáno již v sekci 2.1.4), tak provádím v průběhu rozeznávání jednotlivých objektů filtraci těchto rušivých elementů. Filtr pracuje jednoduše. Rozhoduje podle počtu pixelů patřící objektu, které porovnává s nastavitelnou konstantou, zda se jedná opravdu o srážku či ne.

5.2 Výpočet těžiště

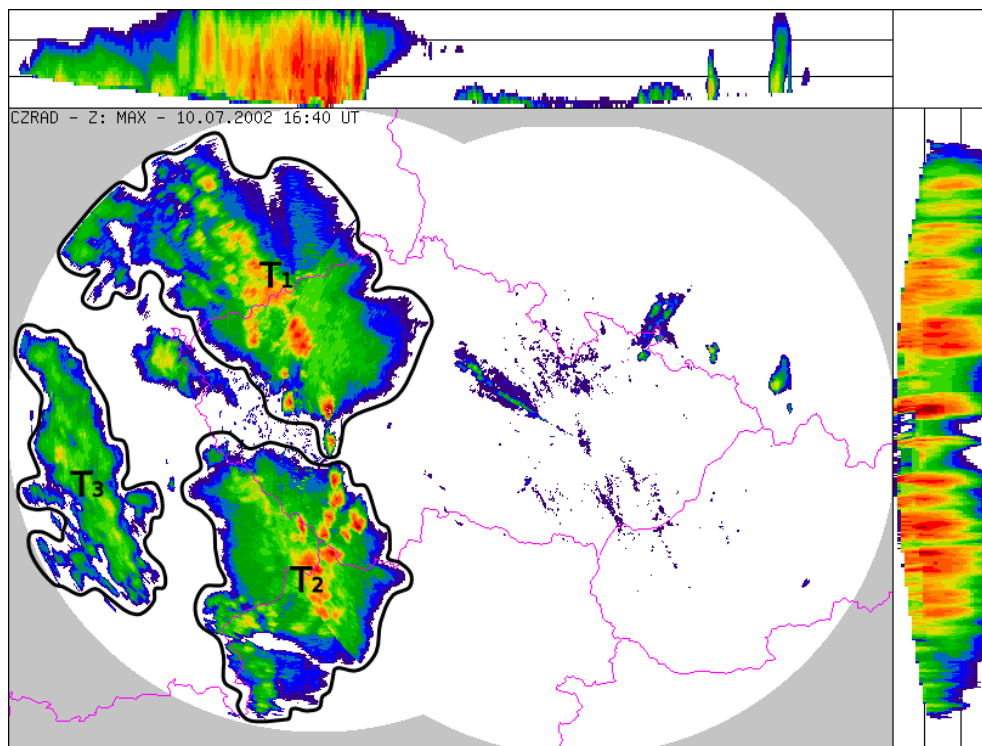
Po extrakci objektů (mraků) máme informaci, které pixely danému mraku náleží včetně jejich barevné informace (RGBA). Na srážkových snímcích mají objekty vlastnost, že na každém dalším snímku změni svůj tvar i počet pixelů (část oblačnosti se vyprší). Proto nemůžeme sledovat polohu oblačnosti podle jednotlivých pixelů, ale musí se vypočítat nějaký referenční bod, který bude určovat polohu celého mraku. Pro výpočet tohoto bodu jsem si zvolil určení těžiště srážkového objektu, které se určí podle vzorce 5.1.

$$x_T = \frac{1}{n} \sum_{i=1}^n x_i \quad y_T = \frac{1}{n} \sum_{i=1}^n y_i \quad (5.1)$$

x_i, y_i – souřadnice bodů jednoho objektu

n – celkový počet bodů objektu

Po výpočtu těžiště pro každý objekt (viz. obrázek 5.1) již není problém zaznamenávat jejich polohu. Protože mraky mění v podstatě ve všech případech svůj objem (počet pixelu), i když by nenastal žádný posun mezi dvoumi statickými snímky, tak při přepočtu



Obrázek 5.1: Ukázka těžišť jednotlivých objektů

těžiště v druhém snímku se toto těžiště vyskytne na jiném místě. Takovou vlastnost, ale můžeme zanedbat, protože mraky mezi jednotlivými snímky mění svůj objem pozvolna (o pár pixelů). Z těchto důvodů nenastávají velké skoky těžišť.

5.3 Problémy mizení a objevování objektů

Srážkové objekty mohou mít vlastnosti mizení a objevování. Náhlý vznik nového mraku bývá méně časté nežli zánik.

Objevení může nastat, když se některá oblačnost rozdělí na dva či více objektů. V takovém případě vznikne nový objekt. Řeším to způsobem, že beru v úvahu pouze jeden objekt. Ostatní, které vznikly musím zanedbat, protože používám Lagrangeův interpolační polynom 2. řádu (jak je popsáno v kapitole 6.4) a ten potřebuje tři vzorky. Vzniklý objekt může poskytnout pouze dva a méně (rozdělení může nastat nejdříve ve druhém snímku a uvažují se vždy tři snímky).

Mizení může nastat pokud se některá srážková oblačnost vyprší nebo dojde ke sloučení dvou či více jednotlivých oblačností. Tento jev ošetřuji přiřazením sloučeného objektu k původnímu největšímu objektu před sloučením a ostatní zanedbám.

5.4 Slučování do větších celků

Abych minimalizoval chyby způsobené pozdějším sloučením nebo rozdělením, zavedl jsem operaci sloučení jednotlivých objektu. Tato operace spočívá ve vyhodnocení do jaké míry

jsou srážky vedle sebe a pokud jsou dostatečně blízko (nastavování konstantou), tak se objekty sloučí v jeden.

Určení vzájemné vzdálenosti objektů by se dalo řešit porovnáváním všech pixelů mezi jednotlivými objekty. Pixely, které jsou sobě nejbližší, určují vzdálenost objektů. Protože objekty často obsahují velký počet pixelů bylo by výkonnostně náročné je vzájemně všechny porovnávat.

Proto jsem zavedl jinou rychlejší metodu, která využívá toho, že polohy mraků sledujeme pomocí jejich těžišť. Nabízí se určit takovouto relativní vzdálenost dvou objektů jako vzdálenost dvou těžišť, ale různorodost tvarů srážkových objektů (viz. 4.1) nedovoluje takové určení.

Řeším to způsobem optimalizace vzdálenosti dvou těžišť vzhledem k jejich objemu. Ten určím jako objem kruhu, kterému se následně spočítá poloměr dle vzorce 5.2. Těmito poloměry se pak kompenzují vzdálenosti pro sloučení dvou objektů (nastaveno konstantou).

$$V_{mraku} = V_{kruhu} = \pi r^2 \implies r = \sqrt{\frac{V_{kruhu}}{\pi}} \quad (5.2)$$

Pro větší názornost uvedeme příklad sloučení dvou objektů, které pochází z reálného snímku. Po odfiltrování pro nás nezájímavých objektů jsme rozpoznali tři srážkové skupiny. Z čehož se dva sloučí v jeden, protože pro to splňují dané podmínky, jak je vidět na obrázku 5.2.

Údaje pro tyto dva slučované objekty (v pixelech) jsou:

1. srážková skupina

- Objem: 34529
- Těžiště: x=207 y=222
- Poloměr: 104

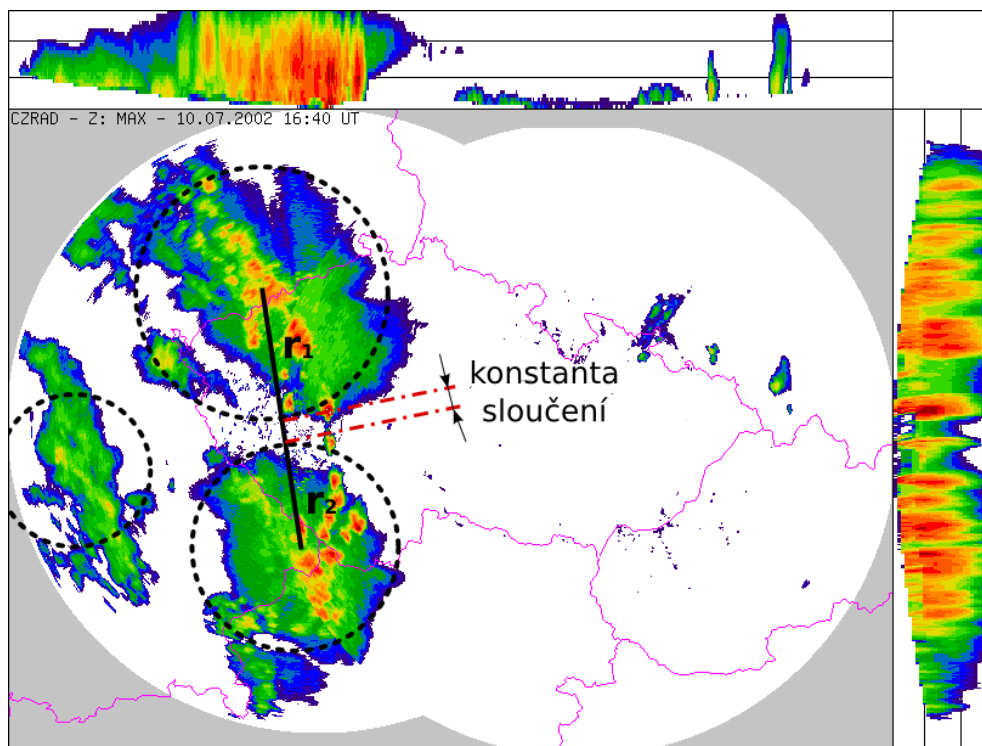
2. srážková skupina

- Objem: 24271
- Těžiště: x=232 y=461
- Poloměr: 87

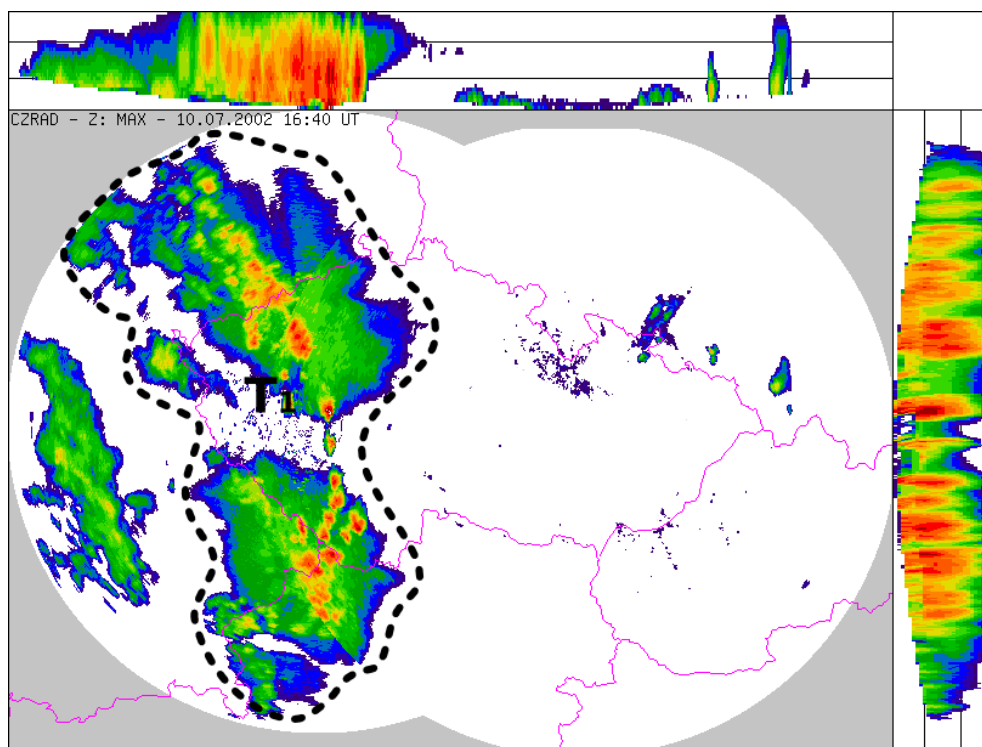
Tyto dva objekty se sloučí v jeden. Sloučení proběhne přiřazením všech pixelů z obou srážek jednomu objektu a následné vypočítání nového těžiště jak je vidět na obrázku 5.3.

5.5 Přiřazení mezi jednotlivými snímky

Po provedení všech optimalizačních kroků, které byly popsány výše, můžeme začít přiřazovat odpovídající objekty na jednotlivých statických snímcích. Parametry rozhodujícího algoritmu párování objektů mezi dvěma snímky byly určeny experimentálně. Tyto parametry určují do jaké míry může nastat určitá změna mezi dvoumi následujícími snímky. Změnami je myšleno např. změna objemu, posun těžiště oblačnosti.



Obrázek 5.2: Znázornění významu konstanty slučování

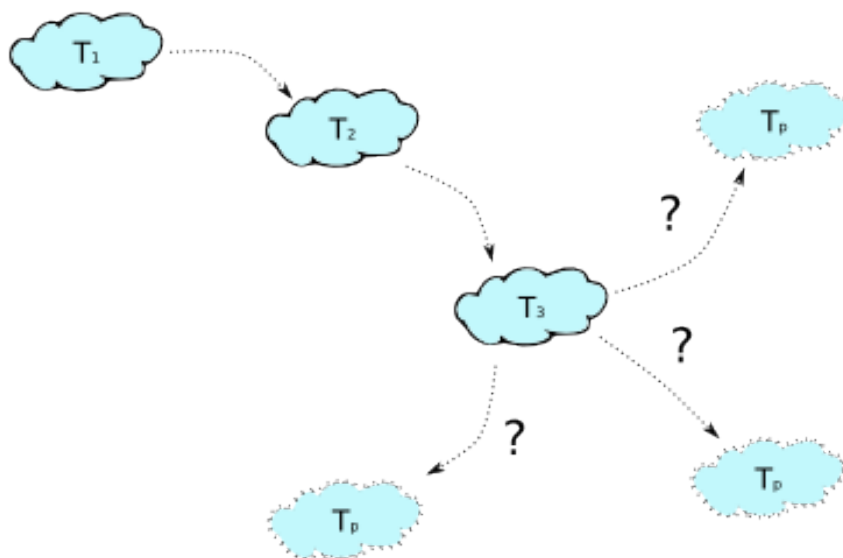


Obrázek 5.3: Znázornění stavu po sloučení

Kapitola 6

Predikce deště

Známe již polohy těžiště jednotlivých mraků na známých snímcích. Dostáváme se tak k dalšímu problému, jak predikovat příští polohy mraků (trend průběhu) jak je znázorněno na obrázku 6.1. Uvažoval jsem dvě metody, které se odvíjejí od lineární a nelineární predikce jak již bylo zmíněno v sekci 2.2.



Obrázek 6.1: Predikce příští polohy mraku

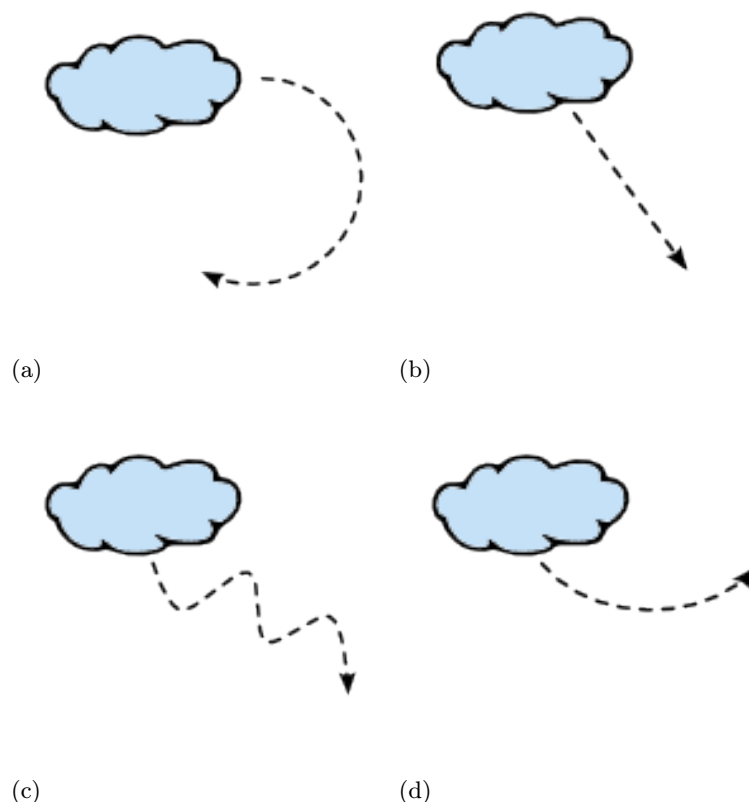
6.1 Lineární predikce

Máme k dispozici dostatečně informací (polohu, rychlost, zrychlení těžiště a čas mezi jednotlivými statickými snímky), tak je možno zjistit příští polohu podle vzorců pro rovnoměrný, rovnoměrně zrychlený či zpomalený pohyb.

Popsaná metoda nevyhovuje našim požadavkům, protože mraky neputují pouze rovnoměrným nebo rovnoměrně zrychleným přímočarým pohybem, ale můžou putovat také po

trajektoriích, které jsou znázorněny na obrázku 8.1.

Pro trajektorie, které nejsou přímočaré, by tato metoda vykazovala velkou chybu predikce.



Obrázek 6.2: Putování mraků - příklad trajektorií

6.2 Nelineární predikce

Nelineární predikce zvládne pracovat i s nepřímými trajektoriemi, které jsou znázorněny na obrázku 8.1 (konkrétně a, c, d).

Jedna z nelineárních predikcí, kterou jsem si vybral, pracuje s interpolací pomocí algebraických polynomů. Tato predikce dokáže vytvořit funkci která popisuje danou trajektorii (přesněji funkci která se trajektorií co nejvíce podobá). Podrobnější vysvětlení viz. 6.3.

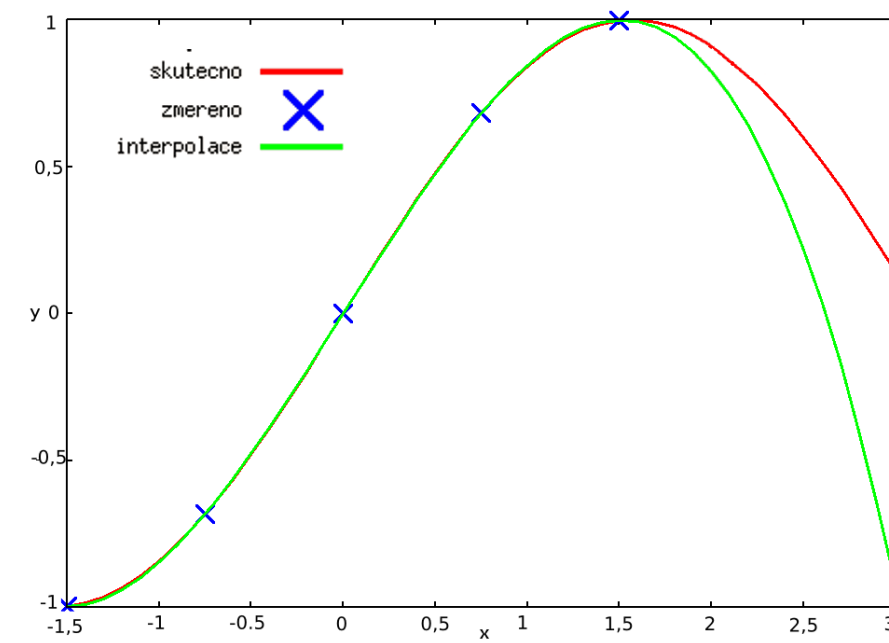
6.3 Interpolace algebraickými polynomy

Dostáváme se k problému jak určit hodnotu funkce. Existují jednoduché funkce, jako je například $f(x) = x^2$, pro které není náročné určování jejich jednotlivých hodnot popřípadě tyto hodnoty najít v příslušných tabulkách.

Pro úlohu predikování pohybu srážek neznáme vhodnou funkci, která by dokázala tento pohyb popsat. Jediné co známe jsou hodnoty tohoto průběhu v určitých bodech (snímky

po deseti minutách).

Jako řešení se nabízí nahradit takovouto funkci některou jinou která se jí určitým způsobem podobá, ale pro potřebné výpočty je vhodnější. Tomuto způsobu se říká interpolace, která je znázorněna na grafu 6.3.



Obrázek 6.3: Interpolace funkce sinus

Pro jednoduchost dalších výpočtů se nejčastěji používá k nahrazení původní funkce algebraický polynom. Náhradní funkci vybíráme podle určitých kritérií. Jako jsou například:

1. Aproximující funkce se má zadaným bodům co nejvíce přiblížit, ale přímo jimi procházet nemusí.
2. Aproximující funkce musí mít s funkcí původní v určitých známých bodech stejné hodnoty.

Druhý případ je vhodnější pro naše účely, kdy známe polohy těžišť jednotlivých mraků v diskrétních časech. Pro interpolaci těchto bodů platí následující teorie.

6.3.1 Princip interpolace

Při interpolaci kdy chceme, aby funkce procházela zadanými body, použijeme určitý interpolační polynom, kterým budeme původní funkci nahrazovat. Pomocí získané funkce můžeme dostat polynom procházející všemi body na intervalu $\langle x_0, x_n \rangle$. Body, které nám jsou na začátku známy a patří původní funkci se nazývají uzly interpolace.

Známe polohy těžišť mraků, kterých je n v různých diskrétních časech. Než se začne vytvářet polynom je potřeba si určit jakého řádu bude. Určuje se to jednoduše a to tak, že známé body nám vytvářejí n interpolačních podmínek. Z toho určíme stupeň polynomu, který je v tomto případě $n - 1$ stupně.

Existuje právě jeden polynom mezi všemi, které jsou nejvíce n -tého řádu a tento jediný polynom je interpolačním polynomem pro původní funkci. Takové tvrzení lze dokázat a to způsobem, který je popsán ve skriptech od RNDr. Břetislava Fajmona a Ph.D., Mgr. Ireny Růžickové [1].

6.3.2 Důkaz

Cituji z více uvedených skript:

Existenci interpolačního polynomu dokážeme tím způsobem, že předvedeme postup, kterým jej lze pro libovolné navzájem různé uzlové body zkonstruovat (viz. 6.4). To, že interpolační polynom procházející danými body existuje právě jeden, dokážeme sporem. Předpokládejme, že existují dva polynomy stupně nanejvýš n , označme je $P_n(x)$ a $R_n(x)$ takové, že $P_n(x_i) = f_i, i = 0, \dots, n$ i $R_n(x_i) = f_i, i = 0, \dots, n$. Ukážeme, že tyto dva polynomy jsou shodné. Za tím účelem označme $Q_n(x) = P_n(x) - R_n(x)$. Je vidět, že $Q_n(x)$ je opět polynom stupně nejvýše n a navíc $Q_n(x_i) = 0, i = 0, \dots, n$. Máme tedy polynom stupně nejvýše n , který má $n + 1$ kořenů. To je možné jedině tak, že $Q_n(x)$ je identicky roven nule, $Q_n(x) \equiv 0$, a tedy $P_n(x) \equiv R_n(x) \forall x \in R$.

6.3.3 Známé metody

Pro určení interpolačního polynomu existuje několik postupů (např. Lagrangeův a Newtonův interpolační polynom). Všechny postupy pro zadanou funkci vytvoří stejný polynom.

Lagrangeův interpolační polynom je jedním ze známějších a také snadných způsobů interpolace funkce zadané pouze v diskrétních bodech.

Newtonův interpolační polynom je pouze modifikace Lagrangeova int. polynomu a výhodu má v tom, že se nemusí při přidání uzlu přepočítávat celý polynom, ale stačí jenom část.

6.4 Lagrangeův interpolační polynom

Pro mé potřeby jsem si vybral Lagrangeův interpolační polynom, který se jeví jako dostačující, protože předem znám počet zkoumaných uzlů (bodů) a v průběhu žádné nejsou přidávány. Hlavním důvodem k výběru této metody byla její schopnost dobře provádět také extrapolaci bodů což je nezbytné pro predikci.

Nejprve bude popsáno jak se konstruuje Lagrangeův interpolační polynom a následně vysvětleno využití zmiňované nezbytné vlastnosti extrapolace pro predikování příštích poloh srážkových objektů.

6.4.1 Konstrukce interpolačního polynomu

Interpolační polynom daný body $[x_i, f_i]$, $i = 0, \dots, n$ sestavíme pomocí polynomů $l_i(x)$ takových, že

$$l_i(x_j) = \begin{cases} 1 & \text{pro } i = j \\ 0 & \text{pro } i \neq j \end{cases}$$

$$l_0(x) = \frac{(x - x_1)(x - x_2) \dots (x - x_n)}{(x_0 - x_1)(x_0 - x_2) \dots (x_0 - x_n)} \quad (6.1)$$

polynom 6.1 má v x_0 hodnotu 1 a v ostatních uzlových bodech hodnotu 0.

Podobně se dají získat i ostatní polynomy l_i , $i = 0, \dots, n$:

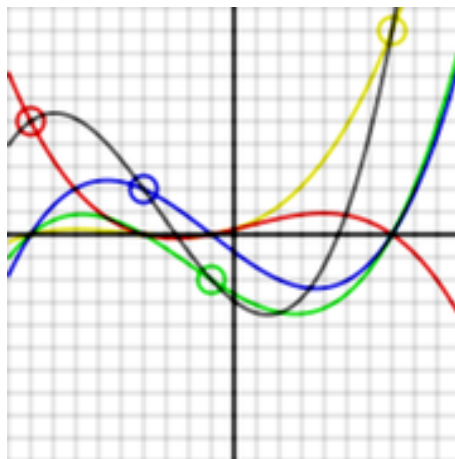
$$l_i(x) = \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_0 - x_1)(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}$$

Interpolační polynom $P_n(x)$ nyní dostaneme jako kombinaci $l_i(x)$:

$$\begin{aligned} P_n(x) &= f_0 l_0(x) + f_1 l_1(x) + \dots + f_n l_n(x) = \\ &= f_0 \frac{(x - x_1)(x - x_2) \dots (x - x_n)}{(x_0 - x_1)(x_0 - x_2) \dots (x_0 - x_n)} + f_1 \frac{(x - x_0)(x - x_2) \dots (x - x_n)}{(x_1 - x_0)(x_1 - x_2) \dots (x_1 - x_n)} + \dots \\ &\quad \dots + f_n \frac{(x - x_0)(x - x_1) \dots (x - x_{n-1})}{(x_n - x_0)(x_n - x_1) \dots (x_n - x_{n-1})} \end{aligned} \quad (6.2)$$

Způsobem jakým bylo ukázáno teoretické odvození Lagrangeova interpolačního polynomu 6.2 bylo převzato ze skript od RNDr. Břetislava Fajmona a Ph.D., Mgr. Ireny Růžičkové [1].

Názorněji si to můžeme popsat jako obrázek 6.4, který byl zveřejněn na wikipedii [2]. Tento obrázek znázorňuje čtyři body $[-9; 5]$, $[-4; 2]$, $[-1; -2]$, $[7; 9]$ proložené (kubickým) interpolačním polynomem $L(x)$, který je součtem pomocných polynomů $y_0l_0(x)$, $y_1l_1(x)$, $y_2l_2(x)$ a $y_3l_3(x)$. Interpolační polynom $L(x)$ prochází všemi zadanými body a pomocné polynomy $l_i(x)$ procházejí svým řídicím bodem $[x_i; y_i]$ a v ostatních bodech $x_j (i \neq j)$ jsou nulové.



Obrázek 6.4: Konstrukce interpolačního polynomu

6.4.2 Příklad

Jako praktickou ukázkou si ukážeme konstrukci interpolačního polynomu, který bude vycházet z reálných čtyř pozic těžišť srážkových objektů. Body jsou čtyři a proto bude polynom stupně třetího.

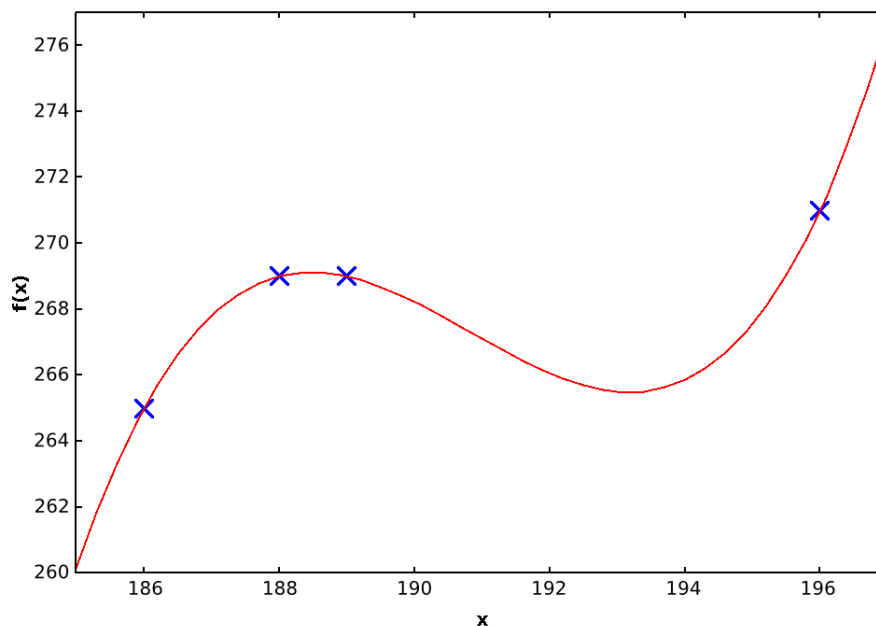
Máme tedy následující body:

x_i	181	186	188	189
f_i	264	265	269	269

Pro konstrukci polynomu použijeme uvedený vzorec 6.2:

$$\begin{aligned}
 P_3(x) = & 264 \frac{(x-186)(x-188)(x-189)}{(181-186)(181-188)(186-189)} + 265 \frac{(x-181)(x-188)(x-189)}{(186-181)(186-188)(186-189)} + \\
 & + 269 \frac{(x-181)(x-186)(x-189)}{(188-181)(188-186)(188-189)} + 269 \frac{(x-181)(x-186)(x-188)}{(189-181)(189-186)(189-188)}
 \end{aligned}$$

Výsledný interpolační polynom je spolu se zadanými body znázorněn na obrázku 6.5.



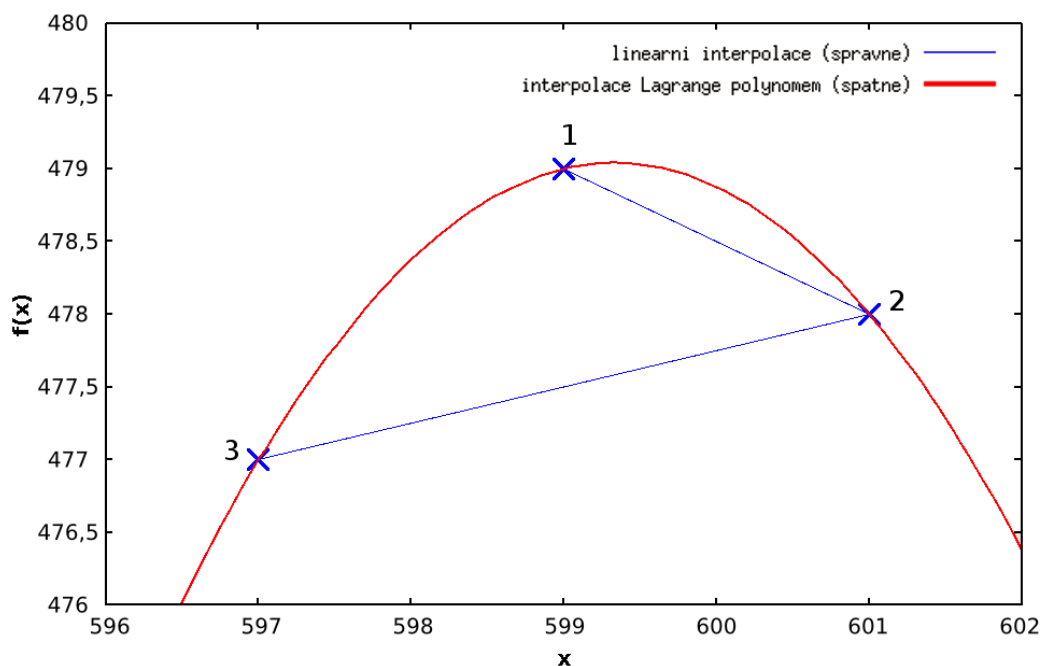
Obrázek 6.5: Zadané body a výsledný interpolační polynom

6.5 Extrapolace využívající Lagrangeovu metodu

Právě pro dobrou schopnost extrapolace jsem si vybral tuto metodu. Predikce by se zdála v podstatě jednoduchá v tom, že po získání interpolačního polynomu, provedeme extrapolaci o známý úsek času. Pro případ predikce deště se vyskytují dva základní problémy. Jeden z nich jsem pracovně pojmenoval „Problém vracení se objektů“ a druhý „Nejpřesnější stupeň polynomu“.

6.5.1 Problém vracení se objektů

Původní myšlenka byla vykreslit trajektorii objektů v XY grafu. Nebo-li na jednu osu zanást souřadnici x a na druhou y . Jak bylo znázorněno například na grafu 6.5. Problém nastal v případě, že pozorovaný objekt začal mít tendence se vracet. Lagrangeův interpolační polynom přestal pracovat jak je vidět na grafu 6.6. Objekt se pohyboval v pořadí bodů 1, 2 a 3. Interpolace funguje správně jen pokud platí $x_{n+1} > x_n$ (x_{n+1} - hodnota souřadnice x na snímku, x_n - hodnota souřadnice x na předchozím snímku).



Obrázek 6.6: Ukázka dat pro, které nepracuje Lagrangeův interpolační polynom

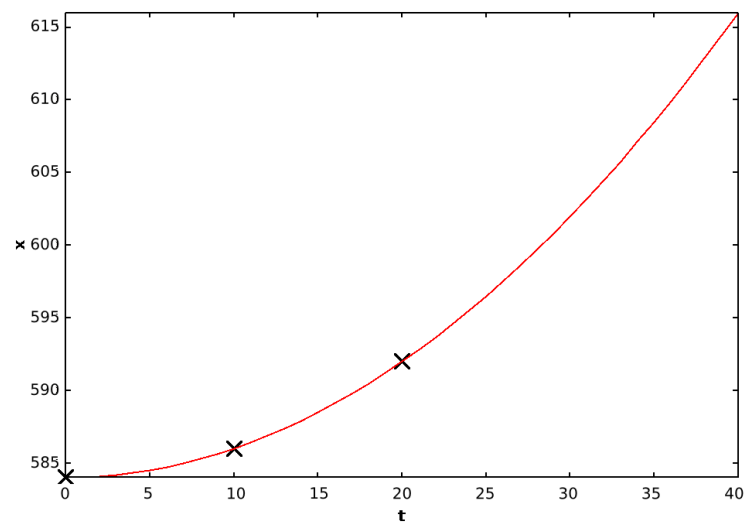
Proto jsem začal používat jinou závislost. Její základní myšlenka je založená na odděleném pozorování souřadnice x a souřadnice y .

Když si vykreslíme zvlášť souřadnici x závislou na čase t a na další graf souřadnici y závislou na stejném čase t , tak nám tento postup s jistotou zaručí, aby nemohlo nastat „vracení se objektu“ (čas nedokáže plynout jinak než dopředu).

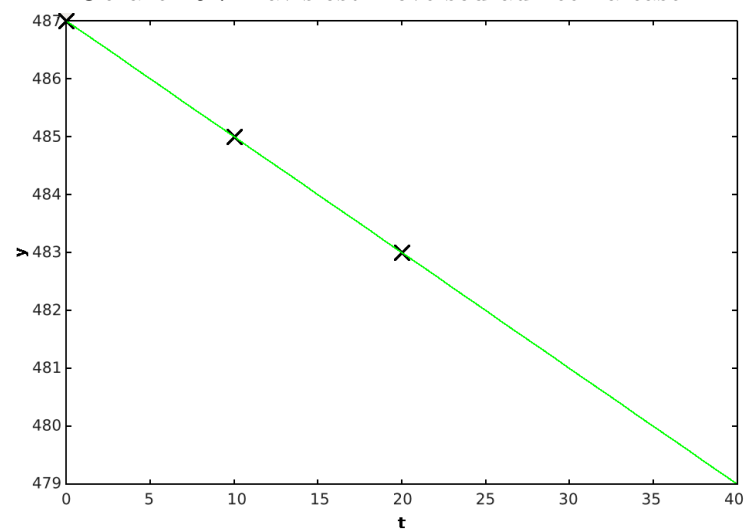
Jakmile známe tyto dva průběhy, můžeme z nich vytvořit výslednou predikovanou pozici, která bude schopna pro jakýkoliv průběh dráhy korektně pracovat.

Predikovaná pozice bude zahrnovat všechny kinematické veličiny pohybu tělesa (poloha, rychlost, zrychlení).

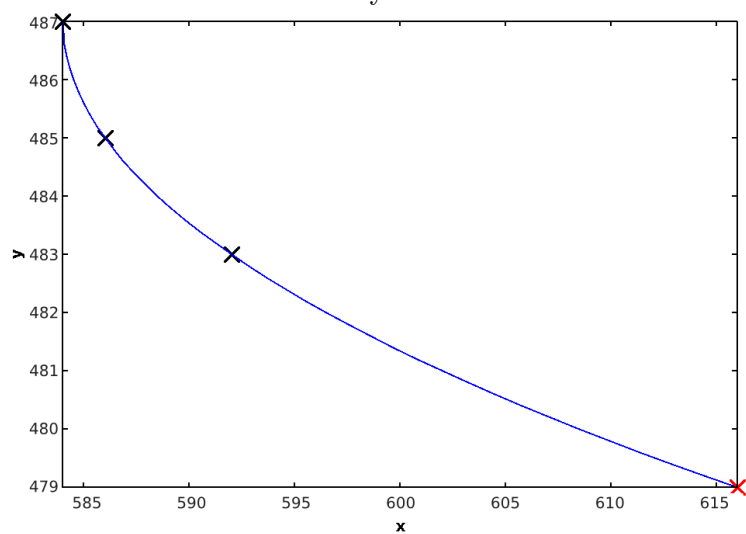
Dobře je to vidět na grafech, kde na prvním 6.7 je znázorněn průběh x v čase a na druhém 6.8 průběh y v tom samém čase. Na grafu 6.9 je výsledný průběh xy kde černé body jsou známé pozice a červený bod je určený predikcí (extrapolací pomocí Lagrangeova interpolačního polynomu).



Obrázek 6.7: Závislost x-ové souřadnice na čase



Obrázek 6.8: Závislost y-ové souřadnice na čase

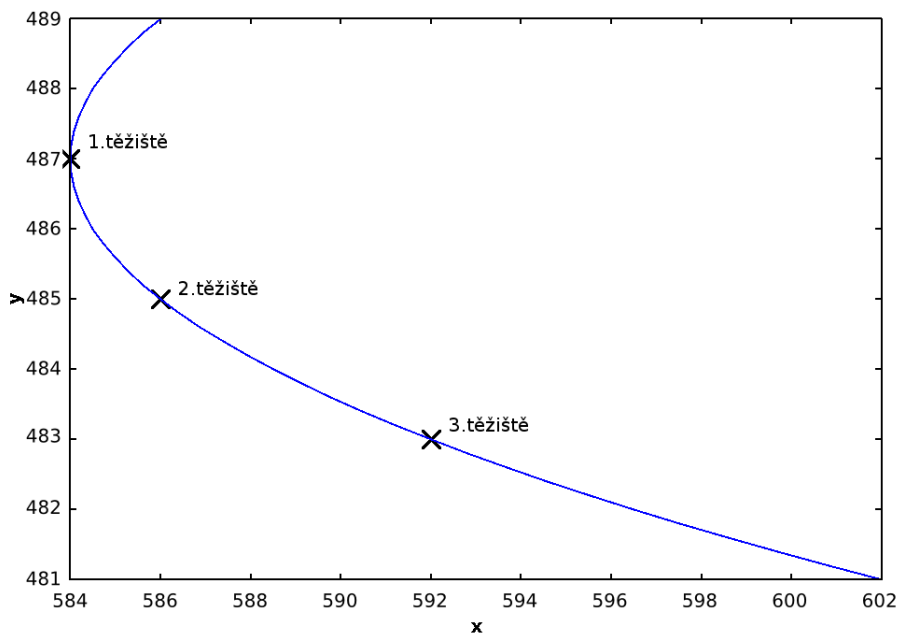


Obrázek 6.9: Výsledná trajektorie

6.5.2 Nejpresnější stupeň polynomu

Další problém, který jsem musel vyřešit je určit si stupeň polynomu, který budu používat při predikci. Snímků máme k dispozici v podstatě neomezeně takže počtem vzorků nejsme omezeni. U Lagrangeova interpolačního polynomu bohužel neexistuje pravidlo, čím více bodu (uzlů) máme k dispozici tím přesnější funkci dostaneme, jak je také zmíněno ve skriptech od RNDr. Břetislava Fajmona a Ph.D., Mgr. Ireny Růžičkové [1]. S příliš velkým počtem uzlů se nám může vyskytovat jev tzv. rozkmitání, který není žádoucí.

Nejpřesnější polynom jsem se rozhodl určit pomocí několika testů, kdy jsem si s využitím open source softwaru Octave vykresloval průběhy interpolací, kde jsem jako vstupní hodnoty používal reálné pozice těžišť oblačností. Vyhodnotil jsem jako nejlepší volbu vycházet ze tří vzorků nebo-li $n + 1$ uzlů \Rightarrow polynom 2. řádu. Polynom 2. řádu určuje tři základní pohyby a to jsou lineární pohyb, pozvolné stáčení doleva a doprava. Mírné otáčení a jeho interpolace/extrapolace při pohybu těžišť oblačnosti je například vidět na obrázku 6.10.



Obrázek 6.10: Interpolace/Extrapolace polynomem 2.řádu

Kapitola 7

Určení polohy predikovaného místa

Abychom mohli určit polohu místa, pro které chceme predikci srážek provádět, tak musíme zavést nějaký polohovací systém.

Máme k dispozici statické snímky srážek nad Českou Republikou. Tyto snímky jsou uloženy v rastrovém formátu PNG. Takže se objevuje problém, jak určit naši aktuální pozici v ČR a přenést ji do daného rastrového formátu.

Mohli bychom zvolit jednoduchou metodu „odhadnutí od oka“. Princip této metody je následující:

1. Podíváme se na statický snímek se srážkami a odhadneme pozici místa, které chceme predikovat (např. určité město).
2. Pomocí libovolného programu pro práci s rastrovými formáty (např. GIMP) zjistíme na jakých souřadnicích ($[x, y]$ počet pixelů) se zvolený objekt nachází.

Tato metoda by mohla být dosti nepřesná a také náročná pro neodborného uživatele aplikace.

Proto jsem zavedl jinou mnohem přesnější a přívětivější metodu, která využívá polohovacího systému GPS. Tento systém je popsán v sekci 7.1.

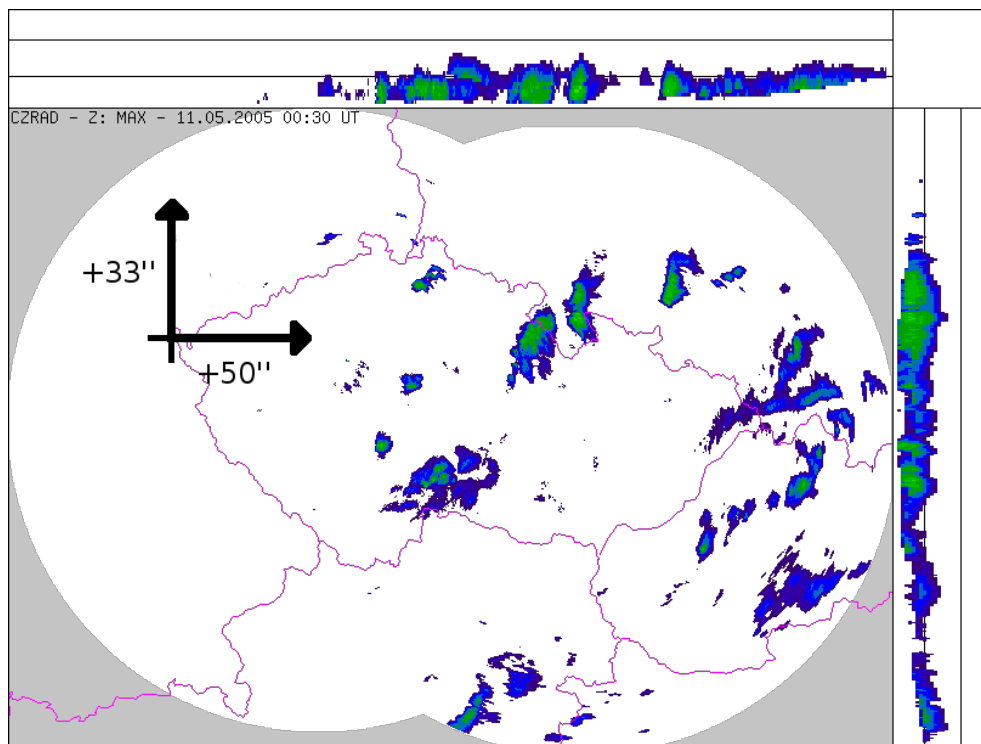
Bohužel nemáme informaci, kde přesně na statickém snímku se nachází místo $[x, y]$ a jeho odpovídající GPS souřadnice. Takže nemáme referenční bod podle kterého bychom synchronizovaly GPS souřadnice s jednotlivými pixely.

Na statickém snímku máme vykresleny hranice ČR. Proto jsem problém převodu GPS souřadnic na formát $[x, y]$ vyřešil následovně.

Pomocí programu pro práci s rastrovými formáty jsem zjistil pozici $[x, y]$ pixelu, který leží na hranici ČR a zároveň je nejvíce na západ jak je znázorněno na obr. 7.1. Tento bod jsem určil jako referenční. Od tohoto bodu se můžeme pohybovat po jednotlivých pixelech přičemž každému jednomu pixelu odpovídá:

- Pohyb v ose x o 1 pixel (1 km) $\rightarrow +55''$
- Pohyb v ose y o 1 pixel (1 km) $\rightarrow +33''$

Pak je již jednoduché převést kteroukoliv GPS souřadnici na formát $[x, y]$ v pixelech.



Obrázek 7.1: Znázornění referenčního bodu

7.1 Souřadnice GPS

GPS (Global Positioning System), je vojenský polohový družicový systém provozovaný Ministerstvem obrany Spojených států amerických, s jehož pomocí je možno určit polohu a přesný čas kdekoli na Zemi nebo nad Zemí s přesností první desítky metrů. Přesnost GPS lze s použitím dalších metod ještě zvýšit až na jednotky centimetrů. Část služeb tohoto systému s omezenou přesností je volně k dispozici i civilním uživatelům.

Vybral jsem si geografické souřadnice (latitude/longitude) na elipsoidu WGS 84. Příklad souřadnic: 49° 11' 42.687" N, 16° 35' 58.88" E.

World Geodetic System 1984 (zkratka WGS 84), česky Světový geodetický systém 1984, je světově uznávaný geodetický standard vydaný ministerstvem obrany USA roku 1984, který definuje souřadnicový systém, referenční elipsoid a geoid pro geodézii a navigaci. V roce 1996 byl rozšířen o zpřesněnou definici geoidu EGM96. Byl vytvořen na základě měření pozemních stanic družicového polohového systému TRANSIT a nahrazuje dřívější systémy WGS 60, WGS 66 a WGS 72.

Souřadnicový systém WGS 84 je pravotočivá kartézská soustava souřadnic se středem v těžišti Země (včetně moří a atmosféry). Kladná osa x směřuje k průsečíku nultého poledníku a rovníku, kladná osa z k severnímu pólu a kladná osa y je na obě předchozí kolmá ve směru doleva (90° východní délky a 0° šířky), tvoří tak pravotočivou soustavu souřadnic.

Přesná a úplná dokumentace souřadnicového systému WGS 84 je na stránkách National Geospatial-Intelligence Agency [4].

Kapitola 8

Vyhodnocení predikce

Predikci jsem otestoval na všech dostupných snímcích. Vytvořil jsem si skript v pythonu, který vypočítá úspěšnosti jednotlivých predikcí. Na doporučení vedoucího práce jsem provedl porovnání výsledků predikcí s kolegou, který pracoval odděleně na stejném tématu bakalářské práce. Porovnání proběhlo na domluveném výběru konkrétních snímků.

8.1 Testování na snímcích

Testovací skript jsem postupně spouštěl pro predikce 20, 40, 80, 100 a 120 minut. Dostupných snímků je velké množství, proto zde nebudu uvádět celé výpisy úspěšností (výpis by zabral mnoho listů). Z těchto testů jsem vybral jeden případ kde lze pozorovat klesání úspěšnosti s rostoucím časem predikce. V některých případech dlouhodobější predikce srážky odputují mimo ČR. Proto výsledný predikovaný snímek (obsahuje pouze ČR) bude bez oblačnosti.

Scored average probability of prediction (rain or not rain)

	+20 [min]	+40 [min]	+80 [min]	+100 [min]	+120 [min]
rating	92.9896 %	89.9821 %	86.37 %	85.2692 %	83.7217 %

8.2 Porovnání řešení

Kolega řešil krátkodobou predikci následujícím způsobem:

K předpovědi deště pomocí neuronových sítí je využito jejich schopností aproximovat nelineární funkce. Na základě předkládání velkého množství vektorů je neuronová síť schopna naučit se základní principy chování dešťových mraků v radarových snímcích. Predikce je potom vytvářena na základě podobnosti s některým již naučeným příkladem.

Domluvili jsme se, že porovnání úspěšnosti aplikace provedeme na snímcích, které byly pořízeny 10.07.2002, kde lze pozorovat putování fronty. Testů proběhlo pět, predikovalo se postupně pro 10, 20, 30, 40 a 50 minut. Reálný snímek pro 10 minut odpovídá času 17:10, pro 20 minut odpovídá času 17:20, atd.

Domluvili jsme se také na stejných metrikách pro vyhodnocování úspěšnosti predikce. Aplikace vždy vytvoří predikovaný a reálný snímek, které porovná. Porovnání probíhá celkem třemi metodami:

1. Uvažují se pouze dva stavy prší a neprší (Scored average probability of prediction (rain or not rain)).
2. Z 15 intenzit přepočítáme snímky na 3 intenzity a ty se pak uvažují při výpočtu úspěšnosti (Scored average probability of correct class prediction).
3. Vypočítává se úspěšnost pro každou ze tří tříd intenzit zvlášť (Class prediction error).

Pro tyto testy jsem si také vytvořil skript v pythonu, který získá od predikující aplikace snímky predikované i reálné. Také zjistí všechny tři způsoby intenzit a uloží je s pomocnými údaji do logovacího souboru. Porovnání je znázorněno v následujících tabulkách a snímcích. Pro moje řešení jsem zvolil název v tabulce „Detection“ a pro kolegovu řešení „NN sites“.

Scored average probability of prediction (rain or not rain)

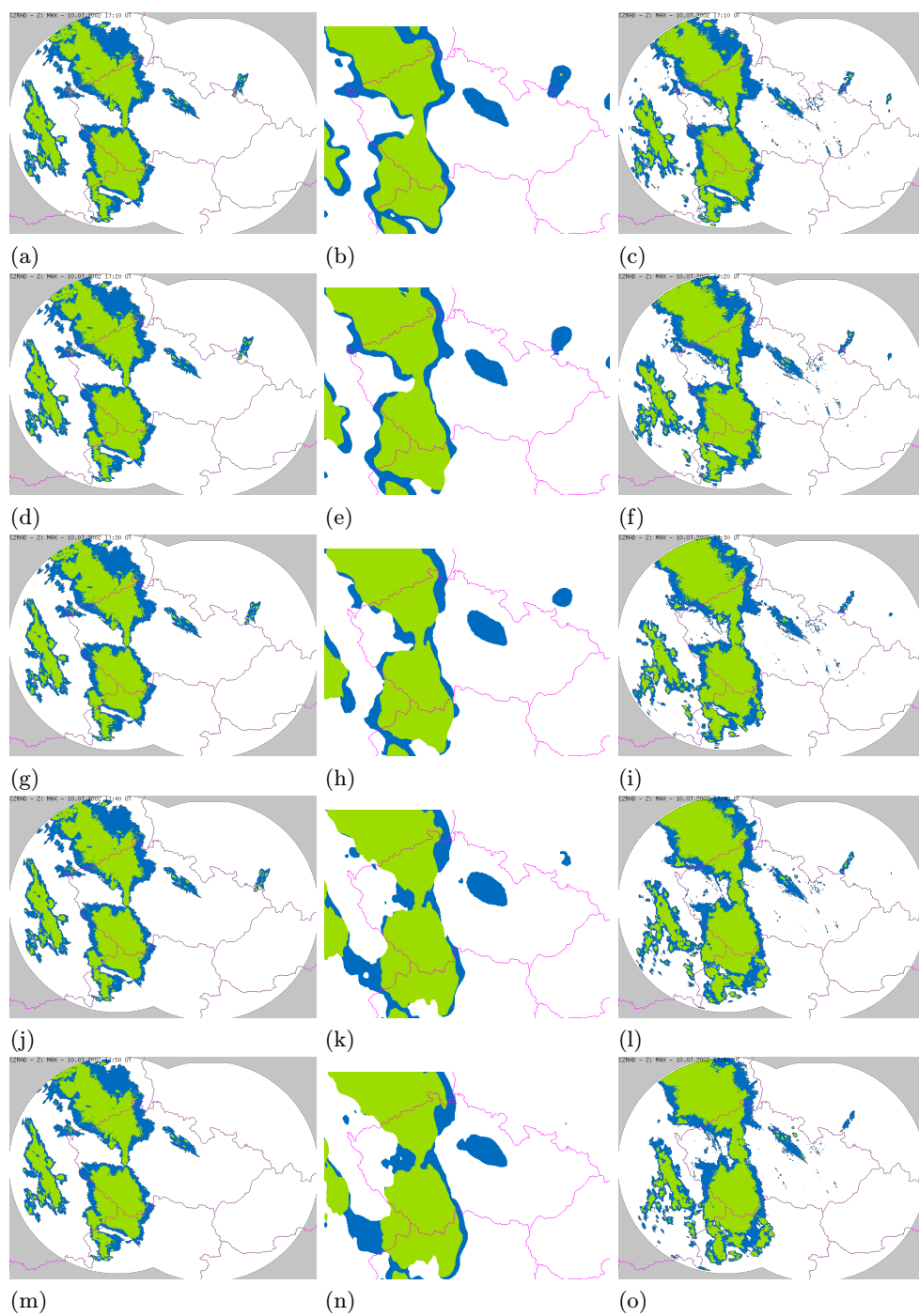
	+10 [min]	+20 [min]	+30 [min]	+40 [min]	+50 [min]
Detection	93.4738 %	91.359 %	89.4754 %	87.6243 %	86.0889 %
NN sites	94.1172 %	93.1586 %	91.7171 %	90.4567 %	86.9939 %

Scored average probability of correct class prediction

	+10 [min]	+20 [min]	+30 [min]	+40 [min]	+50 [min]
Detection	90.7124 %	87.911 %	85.6459 %	83.3865 %	81.7451 %
NN sites	90.5603 %	88.7576 %	87.7864 %	86.3861 %	84.3490 %

Class prediction error

	+10 [min]	+20 [min]	+30 [min]	+40 [min]	+50 [min]
Detection class #0	96.4052 %	94.9847 %	93.6802 %	92.6828 %	91.9892 %
NN sites class #0	95.3122 %	95.1381 %	95.0654 %	94.3111 %	92.9301 %
Detection class #1	61.4923 %	50.4153 %	42.266 %	35.4191 %	29.2212 %
NN sites class #1	55.0865 %	40.6816 %	34.5316 %	29.1617 %	26.0960 %
Detection class #2	73.1513 %	66.1892 %	60.1776 %	53.0993 %	49.1214 %
NN sites class #2	88.9067 %	86.2871 %	84.7304 %	83.5506 %	81.9420 %



Obrázek 8.1: Predikované snímky, 1.sloupec predikce pomocí detekce objektů (a,d,g,j,m), 2.sloupec predikce pomocí NN sítí (b,e,h,k,n), 3.sloupec realita (c,f,i,l,o)

Kapitola 9

Závěr

Před vlastním vývojem aplikace jsem studoval dostupné zdroje pojednávající o detekci pohybu v obraze (případně se zaměřením na meteorologii). Rozhodl jsem pro kombinaci využití algoritmu semínkového vyplňování a extrapolaci kinematiky oblačností pomocí Lagrangeova interpolačního polynomu. Tento přístup nebyl dle získaných zdrojů publikován.

Pro požadovanou krátkodobou predikci (cca +2h) je aplikace funkční. Testoval jsem ji na extrémních datech vybraných ČHMÚ. Je pravděpodobné, že při běžném provozu by byla úspěšnost predikce vyšší.

9.1 Návrhy rozšíření

Aplikace byla od počátku implementována s myšlenkou pozdějšího použití v praxi. A to způsobem, který by využíval aktuálních radarových snímků ČHMÚ (viz. [5]). Proto aplikace počítá s případným rozšířením pro tyto účely.

Nabízí se například rozšíření aplikace o podporu odesílání sms zpráv:

- Uživatel by v sms zprávě uvedl své GPS souřadnice (nebo název města) a aplikace by odpověděla sms zprávou predikci deště.
- Uživatel by obdržel varovnou sms zprávu v případě predikce deště. Podmínky zaslání varování by byly nastavitelné přímo v aplikaci.

Pro větší flexibilitu aplikace by bylo možné rozšíření o podporu více formátů statických snímků. Aktuálně je podporován formát PNG.

Implementovaná podpora GPS systému je funkční. Podporován je systém WGS 84, rozšíření o další systémy je možné.

Další rozšíření by mohlo implementovat grafické uživatelské rozhraní, protože v aktuálním stavu se jedná o konzolovou aplikaci.

Literatura

- [1] Fajmon, B.; Růžicková, I.: Matematika3.
<http://www.umat.feec.vutbr.cz/~fajmon/bma3/matematika3.pdf>, 2005.
- [2] WWW stránky: Lagrange polynomial.
http://www.en.wikipedia.org/wiki/Lagrange_polynomial.
- [3] WWW stránky: Official PNG reference library.
<http://www.libpng.org/pub/png/libpng.html>.
- [4] WWW stránky: Oficiální definice WGS 84, třetí vydání.
http://earth-info.nga.mil/GandG/publications/tr8350.2/tr8350_2.html.
- [5] WWW stránky: Radarové snímky. <http://www.chmi.cz/meteo/rad/index.html>.
- [6] WWW stránky: Remote Sensing. <http://www.remotesensing.org/>.
- [7] WWW stránky: Český hydrometeorologický ústav. <http://www.chmi.cz/>.

Seznam příloh

Příloha A: Manuál

Příloha B: Konfigurační soubor

Příloha C: CD se zdrojovými kódy aplikace i teorie a k nim příslušným dokumentům

Příloha A

Manuál

Program je implementován jako konzolová aplikace (nemá grafické uživatelské rozhraní). Syntaxe spuštění programu se provádí následovně:

```
./rainprediction -[c <city>|gps <gps>] -pt <pred_time> [-at <act_time>]
```

Volby

- c Predikuj pro město s názvem následujícím za přepínačem -c.
- gps Predikuj pro pozici, která je popsána podle systému WGS 84.
- pt Čas pro, který se bude provádět predikce.
- at Nastavení aktuálního času pro testovací účely.

Příklad

```
./rainprediction -gps 49.1952 16.5997 -pt 17:10 -at 17:00
```

nebo,

```
./rainprediction -c Brno -pt 17:10 -at 17:00
```

Na standardní výstup se vypíše informace o postupu operací prováděných aplikací a výsledné predikci:

```
Mesto: Brno
GPS souradnice: 49.1952 16.5997
Souradnice mesta: 458 384
Zadan cas: 17:10
Aktualni cas nastaven: 17:0
Budeme pocitat od snimku s~casem: 16:40
Aktualni cas: 15:24
V~minutach: 1030 - 1000 = 10

Zpracovavam soubor: "/home/salopa/Bakalářská práce/Develop/dev/
images/all/20020710/pacz23.z_max3d.20020710.1640.0.png" .....
```

```
Zpracovavam soubor: "/home/salopa/Bakalářská práce/Develop/dev/
images/all/20020710/pacz23.z_max3d.20020710.1650.0.png" .....
Zpracovavam soubor: "/home/salopa/Bakalářská práce/Develop/dev/
images/all/20020710/pacz23.z_max3d.20020710.1700.0.png" .....
Zpracovavam soubor: "/home/salopa/Bakalářská práce/Develop/dev/
images/all/20020710/pacz23.z_max3d.20020710.1710.0.png" .....
```

POCITAM PREDIKCI...

POZICE TEZIST:

```
-----
[565 271]->[565 265]->[562 260]->[561 256]
[195 330]->[199 326]->[205 323]->[212 323]
-----
```

| Predikce | Realita |

```
-----
teziste: | [556 256] | [561 256] |
polomer mraku: | 20 | 17 |
velikost mraku: | 1281 | 997 |
dest: | 0 | 0 |
-----
```

```
teziste: | [213 321] | [212 323] |
polomer mraku: | 156 | 157 |
velikost mraku: | 77137 | 78358 |
dest: | 0 | 0 |
-----
```

VYTVARIM PREDIKOVANY SNIMEK...

POCITAM USPESNOST...

VYTVARIM VYSLEDNY OBRAZEK...

VYTVARIM VYSLEDNY OBRAZEK...

Scored average probability of prediction (rain or not rain)

RATING_TWO: 92.7851 %

Scored average probability of correct class prediction

RATING_THREE: 89.7328 %

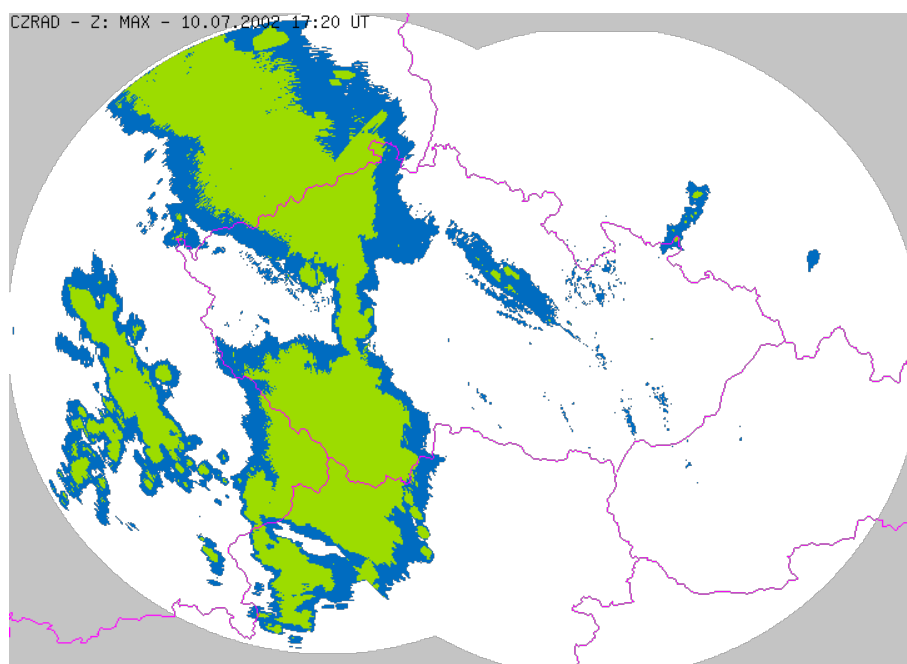
Class prediction error

RATING_THREE_ONE: 95.8915 %

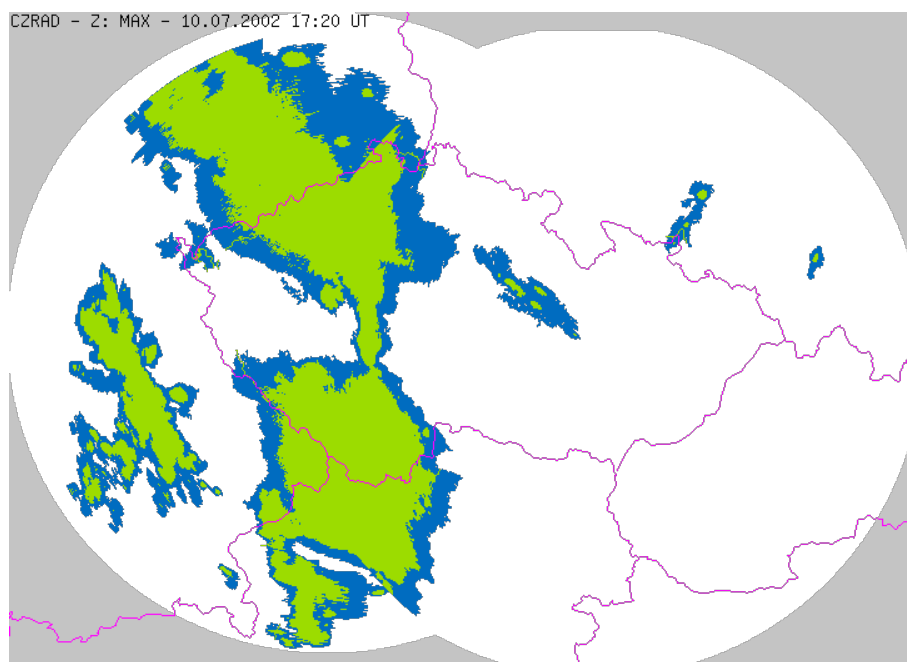
RATING_THREE_TWO: 57.9888 %

RATING_THREE_THREE: 70.8544 %

Jako výstup se také vygenerují dva png obrázky, kde bude vykreslen snímek s reálnými srážkami a s predikovanými srážkami v požadovaný čas:



Obrázek A.1: Výsledný reálný snímek



Obrázek A.2: Výsledný predikovaný snímek

Příloha B

Konfigurační soubor

Aplikace obsahuje také konfigurační soubor, kde se dají nastavit různé parametry ladící algoritmus detekce, registrace atd. Nebo se zde dá také nastavit např. název výstupního snímku predikce, reality atd.

Nastavení rozměrů obrázku a rozsah, který se z obrázku bude uvažovat (pomyslný obdélník)

```
// Sirka a vyska snimku png
const int WIDTH = 810;
const int HEIGHT = 610;

// Rozsah identifikace jednotlivych objektu (mraku)
const int RANGE_X_BEG = 1, RANGE_Y_BEG = 82;
const int RANGE_X_END = 727, RANGE_Y_END = 608;
```

Nastavení názvu výstupního snímku predikce a reality

```
// Nazev souboru kde se vykresli predikovany snimek
const string OUT_FILE = "predikce.png";
const string OUT_FILE2 = "realita.png";
```

Počet snímků z kterých se určuje predikce

```
// Pocet snimku, z kterych se bude odvozovat predikce
const int NUM_OF_IMG = 3;
```

Nastavování parametrů algoritmů pro filtraci rušivých elementů, slučování objektů a také pro registraci objektů mezi jednotlivými snímky

```
// Urceni velikosti mraku ktery se ma zanedbavat
const unsigned int SMALL_CLOUD = 100;

// Nastaveni maximalni vzdalenosti dvou tezist mraku aby doslo k sloucení
const int MERGE_CLOUDS = 100;

// Nastaveni maximalniho rozdilu velikosti mraku
```



```
// 25% - jednotky jsou procenta
const float CLOUD_SIZE_DIFF = 25;
// Nastavení maximální velikosti vektoru pohybu mraku
const float CLOUD_VECTOR_SIZE = 30;
```

Nastavení cesty, kde jsou uloženy snímky s kterými se bude pracovat

```
// Cesta ke vstupním statickým snímkům
const string IMAGES_DIR = "/home/salopa/Bakalářská práce/Develop/dev/
images/all/20020710/pacz23.z_max3d.";
```

Nastavení aktuálního data a času, aby jsme mohli provádět testy i na snímcích z minulosti

```
// Nastavení aktuálního data a času

// 16.04.2006
const string DATE_YEAR = "2002";
const string DATE_MONTH = "07";
const string DATE_DAY = "10";

// 13:20 = 800 minut
const int ACTUAL_TIME_H = 17;
const int ACTUAL_TIME_M = 20;
```

Zde jsou konstanty používaných barev

```
// Konstanty barev vyskytujících se ve snímku z radaru
const unsigned int RGBA_BLACK[] = {0,0,0,255}; //000000ff
const unsigned int RGBA_WHITE[] = {255,255,255,255}; //ffffff
const unsigned int RGBA_MARGINS[] = {196,196,196,255}; //c4c4c4ff
const unsigned int RGBA_DATE[] = {0,0,0,255}; //000000ff
const unsigned int RGBA_BORDERLINE[] = {255,0,255,255}; //ff00ffff

// Barevná paleta - pouze tyto barvy se mohou vyskytovat v mraku
// Třetí intenzita
const unsigned int RGBA_CM_WHITE[] = {252,252,252,255}; //fcfcfcff
const unsigned int RGBA_CM_DRED[] = {160,0,0,255}; //a00000ff
const unsigned int RGBA_CM_RED[] = {252,0,0,255}; //fc0000ff
const unsigned int RGBA_CM_LRED[] = {252,88,0,255}; //fc5800ff
const unsigned int RGBA_CM_ORANGE[] = {252,132,0,255}; //fc8400ff
const unsigned int RGBA_CM_LORANGE[] = {252,176,0,255}; //fcb000ff
const unsigned int RGBA_CM_YELLOW[] = {224,220,0,255}; //e0dc00ff
const unsigned int RGBA_CM_YGREEN[] = {156,222,0,255}; //9cde00ff
const unsigned int RGBA_CM_LGREEN[] = {52,216,0,255}; //34d800ff
const unsigned int RGBA_CM_GREEN[] = {0,188,0,255}; //00bc00ff
const unsigned int RGBA_CM_DGREEN[] = {0,160,0,255}; //00a000ff
// Třetí intenzita
// Druhá intenzita
```

```

const unsigned int RGBA_CM_LBLUE[] = {0,108,192,255}; //006cc0ff
const unsigned int RGBA_CM_BLUE[] = {0,0,252,255}; //0000fcff
const unsigned int RGBA_CM_LMAGENTA[] = {48,0,168,255}; //3000a8ff
const unsigned int RGBA_CM_MAGENTA[] = {56,0,112,255}; //380070ff
// Druha intenzita
// Prvni intenzita
const unsigned int RGBA_BACKGROUND[] = {0,0,0,0}; //ffffffff
// Prvni intenzita

// Barvy vyslednych intenzit
const unsigned int RGBA_PRED_BLUE[] = {0,108,192,255}; //006cc0ff
const unsigned int RGBA_PRED_GREEN[] = {156,220,0,255}; //9cdc00ff

```

Hodnoty pro systém GPS souřadnic

```

// Prevodni konstanty pro sekundy na pixely
const float lat_one_pix_sec = 33; // 1px=33''
const float lon_one_pix_sec = 50; // 1px=50''

const gps offset_gps = {50.2489, 12.0958};
const city offset_xy = {134, 270};

// Konstanty podporovanych mest a jejich souradnic
const int NUM_CITIES = 4;
const string CITIES[NUM_CITIES] = {"brno", "zlin", "zlín", "praha"};

const city BRNO = {453, 391}; // x,y - v~pixelech
const city ZLIN = {0, 0}; // x,y - v~pixelech
const city PRAHA = {0, 0}; // x,y - v~pixelech

```